

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-105654

(43)Date of publication of application : 11.04.2000

(51)Int.Cl. G06F 3/00

(21)Application number : 10-288679 (71)Applicant : CANON INC

(22)Date of filing : 28.09.1998 (72)Inventor : SHIBAO KOKI
WATANABE HIROYASU
SASAKI SEIJI

(54) MULTIWINDOW SYSTEM, SCREEN TRANSITION METHOD, AND STORAGE
MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a multiwindow system capable of efficiently executing a screen transition including a nesting flow and a sequential flow.

SOLUTION: A multiwindow program 1 is provided with a screen transition table 1d. The screen transition table 1d is provided with a hierarchical structure table describing the hierarchical structure of dialog box groups to be transited so as to overlap, and a transition destination table describing the corresponding relation between an event inputted from a user through an input device 4 and the dialog box to become the transition destination of the dialog box to this inputted event. Based on the screen transition table 1d, a screen transition control part 1b retrieves the dialog box at the transition destination so as to automatically execute the screen transition between the

dialog boxes and a screen display part 1c performs control so as to display the retrieved dialog box on the display means 6.

LEGAL STATUS [Date of request for examination] 25.11.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.*** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the multi-window system which displays the screen of dialogic operation with a user on a display means The table which described the layered structure of the dialog group which changes so that it may lap, The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the user and the this inputted event, The multi-window system characterized by having the screen transition control means controlled to perform screen transition between said dialogs based on said screen transition table.

[Claim 2] Said screen transition table is a multi-window system according to claim 1 characterized by it being possible to define two or more transition places to a modal dialog which carries out a multiple-times appearance using the same resource.

[Claim 3] Said screen transition table is a multi-window system according to

claim 1 which carries out grouping of the transition of two or more dialogs to one collectively, and is characterized by the ability to describe per group.

[Claim 4] Said screen transition table is a multi-window system according to claim 1 characterized by the ability to change the transition place to the event of the user of the dialog by the side of [which was shared] it according to the transition situation till then to the dialog and dialog group to whom share-ization was given.

[Claim 5] Furthermore, the multi-window system according to claim 1 characterized by having a cancellation check means to check cancellation of this dialog, and a notice means to notify cancellation of this dialog to the owner window of this dialog when cancellation of said dialog is checked by said cancellation check means when a dialog is canceled according to said screen transition table.

[Claim 6] Furthermore, the multi-window system according to claim 1 characterized by having a cancellation means to cancel this dialog, without performing screen transition to said user's event which displayed the specific dialog which disregarded said screen transition table and the user set up, and was inputted on this specific dialog.

[Claim 7] In the screen transition approach used for the multi-window system which displays the screen of dialogic operation with a user on a display means

The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the table and user who described the layered structure of the dialog group which changes so that it may lap, and the this inputted event is created.

The screen transition approach characterized by controlling to perform screen transition between said dialogs based on said screen transition table.

[Claim 8] Said screen transition table is the screen transition approach according to claim 7 characterized by it being possible to define two or more transition places to a modal dialog which carries out a multiple-times appearance using the same resource.

[Claim 9] Said screen transition table is the screen transition approach according to claim 7 which carries out grouping of the transition of two or more dialogs to one collectively, and is characterized by the ability to describe per group.

[Claim 10] Said screen transition table is the screen transition approach according to claim 7 characterized by the ability to change the transition place to the event of the user of the dialog by the side of [which was shared] it according

to the transition situation till then to the dialog and dialog group to whom share-ization was given.

[Claim 11] Furthermore, the screen transition approach according to claim 7 characterized by notifying cancellation of this dialog to the owner window of this dialog when a dialog is canceled according to said screen transition table, cancellation of this dialog is checked and cancellation of said dialog is checked.

[Claim 12] Furthermore, the screen transition approach according to claim 7 characterized by canceling this dialog, without performing screen transition to said user's event which displayed the specific dialog which disregarded said screen transition table and the user set up, and was inputted on this specific dialog.

[Claim 13] In the storage which stored the program for building the multi-window system which displays the screen of dialogic operation with a user on a display means said program The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the table and user who described the layered structure of the dialog group which changes so that it may lap, and the this inputted event, The storage characterized by including the screen transition

control module controlled to perform screen transition between said dialogs based on said screen transition table.

[Claim 14] Said screen transition table is a storage according to claim 13 characterized by it being possible to define two or more transition places to a modal dialog which carries out a multiple-times appearance using the same resource.

[Claim 15] Said screen transition table is a storage according to claim 13 which carries out grouping of the transition of two or more dialogs to one collectively, and is characterized by the ability to describe per group.

[Claim 16] Said screen transition table is a storage according to claim 13 characterized by the ability to change the transition place to the event of the user of the dialog by the side of [which was shared] it according to the transition situation till then to the dialog and dialog group to whom share-ization was given.

[Claim 17] Said program is a storage according to claim 13 further characterized by including the cancellation check module which checks cancellation of this dialog, and the notice module which notifies cancellation of this dialog to the owner window of this dialog when cancellation of said dialog is checked with said cancellation check module when a dialog is canceled according to said

screen transition table.

[Claim 18] Said program is a storage according to claim 13 characterized by including the cancellation module which cancels this dialog, without performing screen transition to said user's event which displayed the specific dialog which disregarded said screen transition table and the user set up further, and was inputted on this specific dialog.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the screen transition approach and storage which are used for a multi-window system and it.

[0002]

[Description of the Prior Art] In OA equipment, such as a copying machine, etc., the control unit for usually inputting the set point about the setup instruction in the mode for which it asks, and this mode etc. is prepared. As current and this control unit, an alter operation screen is displayed on display means, such as a liquid crystal display panel, and it is constituting in use so that the above-mentioned input may be performed by dialogic operation using this alter operation screen. A setup is advanced, while there is much what is constituted so that an alter operation screen may be made to change when the multi-window system which is used for this control unit, and which generated and displays an alter operation screen changes between two or more dialogs and it repeats

generation of a dialog, and cancellation on a control unit by the user.

[0003] It explains referring to drawing 37 thru/or drawing 42 about the screen transition approach used for such a windowing system. Drawing showing the example of screen transition by the nesting flow whose drawing 37 is one of the screen transition approaches of a multi-window system, Drawing showing the tree structure of the window [drawing 38] in the case of a nesting flow, Drawing showing the example of screen transition by the sequential flow whose drawing 39 is other one of the screen transition approaches of a multi-window system, Drawing showing the tree structure of a window in case drawing 40 generates the screen transition by the sequential flow, drawing in which drawing 41 shows the example of screen transition for which a dialog only disappears, and drawing 42 are drawings showing an example showing screen transition of a window layered structure.

[0004] First, if a nesting flow is explained, as shown in drawing 37 , a nesting flow will change so that a dialog may lap on it by the event (generally depression of the carbon button on a dialog) from a user. If a carbon button is pushed on a dialog 1, specifically, the dialog 2 corresponding to this pushed carbon button will be displayed in piles on a dialog 1. When such transition takes place, the

possession relation of a window is expressed with the tree structure shown in drawing 38 .

[0005] On the other hand, as shown in drawing 39 , a sequential flow changes so that a dialog may interchange by the event from a user. If an event is specifically inputted from a user on the dialog 1 currently displayed on the forefront side, the dialog 2 corresponding to the carbon button with which the dialog 1 was once canceled and was pushed will be displayed (if a carbon button is pushed). The possession relation of the window in this transition is shown by the tree structure shown in drawing 40 , and a dialog 1 and a dialog 2 have it to the same hierarchy.

[0006] Moreover, as shown in drawing 41 , there is also transition to which the dialog currently displayed only disappears.

[0007] The combination of the above-mentioned transition can express screen transition of the application using a dialog fundamentally, the window layered structure is acquired by combining the structure shown in drawing 38 and drawing 40 , and the structure consists of structure as shown in drawing 42 .

Usually, two or more carbon buttons are displayed on one screen, and the screen or actuation which changes according to the carbon button which the user pushed changes. For example, in the layered structure shown in drawing

42 , if the "upper" carbon button is pushed when the dialog 1 is displayed, a dialog 2 will be displayed that a dialog 3 laps with the front face of a dialog 1, when a "lower" carbon button is pushed.

[0008] Although it was common that hard coding was carried out within the code of application as for such screen transition actuation In order to enable screen transition control using reuse and the tool of an application code Separation and the need of controlling screen transition by table-izing and referring to it produce the information on screen transition from application. The technique which replacing a screen one after another automatically according to the event from a user makes possible is already proposed using this screen transition table. As this technique proposed, there are some which are indicated by JP,6-168203,A, JP,9-097158,A, etc. Specifically, the automation technique of the transition which replaces the screen in order is indicated by both.

[0009] Moreover, the screen transition approach which simplifies the procedure of making a deep hierarchy's dialog changing is shown in JP,8-016350,A.

[0010]

[Problem(s) to be Solved by the Invention] However, transition which changes the screen to JP,6-168203,A and JP,9-097158,A in order with the technique of a

publication cannot be automated, and screen transition it is displayed that accumulates a dialog on a front face cannot be performed automatically. That is, the format of the screen transition table for performing automatically screen transition it is displayed that accumulates a dialog on a front face is not established.

[0011] Moreover, a screen change which changes automatically in the dialog which does not have direct possession relation in it about a nesting flow and a sequential flow using a screen transition table since the possession relation between dialogs is not taken into consideration by it while there is no concept of the nesting flow mentioned above in the screen transition approach given in JP,8-016350,A cannot be performed.

[0012] The purpose of this invention is to offer the multi-window system, the screen transition approach, and storage which can perform efficiently screen transition including a nesting flow and a sequential flow.

[0013]

[Means for Solving the Problem] In the multi-window system by which invention according to claim 1 displays the screen of dialogic operation with a user on a display means The table which described the layered structure of the dialog

group which changes so that it may lap, The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the user and the this inputted event, It is characterized by having the screen transition control means controlled to perform screen transition between said dialogs based on said screen transition table.

[0014] Invention according to claim 2 is characterized by the ability of said screen transition table to define two or more transition places to a modal dialog which carries out a multiple-times appearance using the same resource in a multi-window system according to claim 1.

[0015] In a multi-window system according to claim 1, said screen transition table carries out grouping of the transition of two or more dialogs to one collectively, and invention according to claim 3 is characterized by the ability to describe per group.

[0016] Said screen transition table is characterized by the ability of invention according to claim 4 to change the transition place to the event of the user of the dialog by the side of [which was shared] it in a multi-window system according to claim 1 to the dialog and dialog group to whom share-ization was given

according to the transition situation till then.

[0017] In a multi-window system according to claim 1, further, invention according to claim 5 is characterized by having a cancellation check means to check cancellation of this dialog, and a notice means to notify cancellation of this dialog to the owner window of this dialog when cancellation of said dialog is checked by said cancellation check means, when a dialog is canceled according to said screen transition table.

[0018] In a multi-window system according to claim 1, invention according to claim 6 displays the specific dialog which disregarded said screen transition table and the user set up further, and it is characterized by having a cancellation means to cancel this dialog, without performing screen transition to said user's event inputted on this specific dialog.

[0019] In the screen transition approach used for the multi-window system by which invention according to claim 7 displays the screen of dialogic operation with a user on a display means The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the table and user who described the layered structure of the dialog group which changes so that it may lap, and the

this inputted event is created. It is characterized by controlling to perform screen transition between said dialogs based on said screen transition table.

[0020] Invention according to claim 8 is characterized by the ability of said screen transition table to define two or more transition places to a modal dialog which carries out a multiple-times appearance using the same resource in the screen transition approach according to claim 7.

[0021] In the screen transition approach according to claim 7, said screen transition table carries out grouping of the transition of two or more dialogs to one collectively, and invention according to claim 9 is characterized by the ability to describe per group.

[0022] Said screen transition table is characterized by the ability of invention according to claim 10 to change the transition place to the event of the user of the dialog by the side of [which was shared] it in the screen transition approach according to claim 7 to the dialog and dialog group to whom share-ization was given according to the transition situation till then.

[0023] In the screen transition approach according to claim 7, invention according to claim 11 is characterized by notifying cancellation of this dialog to the owner window of this dialog, when a dialog is canceled according to said

screen transition table, cancellation of this dialog is checked and cancellation of said dialog is checked further.

[0024] In the screen transition approach according to claim 7, invention according to claim 12 displays the specific dialog which disregarded said screen transition table and the user set up further, and it is characterized by canceling this dialog, without performing screen transition to said user's event inputted on this specific dialog.

[0025] In the storage which stored the program for invention according to claim 13 to build the multi-window system which displays the screen of dialogic operation with a user on a display means The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the table and user who described the layered structure of the dialog group which changes so that said program may lap, and the this inputted event, It is characterized by including the screen transition control module controlled to perform screen transition between said dialogs based on said screen transition table.

[0026] Invention according to claim 14 is characterized by the ability of said screen transition table to define two or more transition places to a modal dialog

which carries out a multiple-times appearance using the same resource in a storage according to claim 13.

[0027] In a storage according to claim 13, said screen transition table carries out grouping of the transition of two or more dialogs to one collectively, and invention according to claim 15 is characterized by the ability to describe per group.

[0028] Said screen transition table is characterized by the ability of invention according to claim 16 to change the transition place to the event of the user of the dialog by the side of [which was shared] it in a storage according to claim 13 to the dialog and dialog group to whom share-ization was given according to the transition situation till then.

[0029] Invention according to claim 17 is characterized by said program containing further, the cancellation check module which checks cancellation of this dialog, and the notice module which notifies cancellation of this dialog to the owner window of this dialog when cancellation of said dialog is checked with said cancellation check module, when a dialog is canceled according to said screen transition table in a storage according to claim 13.

[0030] Invention according to claim 18 is characterized by said program

containing the cancellation module which cancels this dialog, without performing screen transition to said user's event which displayed the specific dialog which disregarded said screen transition table and the user set up further, and was inputted on this specific dialog in a storage according to claim 13.

[0031]

[Embodiment of the Invention] It explains referring to drawing about the gestalt of operation of this invention below.

[0032] Drawing 1 is the block diagram showing the configuration of the information processor for building one gestalt of operation of the multi-window system of this invention.

[0033] An information processor is equipped with CPU3 which builds a multi-window system based on the multi-window program 1 stored in memory 5 while it controls the whole equipment according to OS2 to be shown in drawing 1 . The multi-window program 1 contains screen information attaching part 1a holding screen information, screen transition control-section 1b, screen-display section 1c, and screen transition table 1d. The transition place table (shown in drawing 9) TA 2 which described correspondence relation with the dialog used as the layered structure table (TA1 shown in drawing 9) which described the

layered structure of the dialog group which changes so that it may lap, and the transition place of the dialog of the event inputted through the input device 4 from the user and the this inputted event is contained in screen transition table 1d. Screen transition control-section 1b searches the dialog of a transition place so that screen transition between dialogs may be automatically performed based on screen transition table 1d, and screen-display section 1c is controlled to display the searched dialog on the display means 6.

[0034] Next, it explains, referring to drawing 2 about the screen transition control in the gestalt of this operation. Drawing 2 is the conceptual diagram showing the configuration for realizing screen transition control in the multi window currently built on the information processor of drawing 1 .

[0035] With the gestalt of this operation, three external public presentation functions were prepared as a function required in order to perform screen transition automatically, and after creating a screen transition table, automation of screen transition is realized by calling each above-mentioned function within application.

[0036] Specifically, three functions, the function which starts screen transition, the function which makes a screen change, and the function which the layered

structure described by the screen transition table is disregarded [function], and displays a specific dialog on a front face most, are held as the above-mentioned external public presentation function at the screen transition engine. If an event is inputted from a user on a dialog (screen), the external public presentation function corresponding to the this inputted event will be called, and with a screen transition engine, creation of a dialog and cancellation will be performed so that screen transition according to the function called with reference to the screen transition table may be performed.

[0037] Next, it explains, referring to drawing 3 about the procedure of screen transition. Drawing 3 is a flow chart which shows the procedure of the screen transition in the multi-window system currently built on the information processor of drawing 1 .

[0038] If a carbon button is pushed on a standard screen, as shown in drawing 3 , the screen which should be first displayed on a degree with the pushed carbon button with reference to a screen transition table in step S31 will be searched. Subsequently, if the carbon button judged and pushed [whether the carbon button progressed and pushed on step S32 is a termination carbon button and] is a termination carbon button, it will return to a standard screen. If the pushed

carbon button is not a termination carbon button, it will be displayed that progress to step S33, a screen is canceled, and it displays in piles, or a screen is exchanged. Subsequently, the screen corresponding to the carbon button again pushed on step S31 on return and the screen currently displayed is searched. And if a termination carbon button is pushed on the screen currently displayed and other carbon buttons other than return and a termination carbon button will be pushed on a standard screen, screen transition corresponding to this pushed carbon button will be performed.

[0039] Next, it explains concretely, referring to drawing 4 about the procedure of screen transition initiation. Drawing 4 is drawing showing the procedure of the screen transition initiation in the multi-window system of the information processor of drawing 1 .

[0040] In case screen transition is started, first, a carbon button is pushed on a dialog and the function with which an event starts the screen transition generated and mentioned above is called. With reference to a screen transition table, screen transition is started by call of this function, the dialog started by the default within that group is created, and it expresses as a screen transition engine on a screen. Here, in performing screen transition newly while other

screen transition already having been performed, all the already developed dialogs are canceled, and it ends, and starts new screen transition after that.

[0041] Once this screen transition is started, according to the contents of the screen transition table, transition will be repeated to the event inputted by the user. In making a screen change to the event inputted by the user, it calls the function which makes a screen change in the event handler. Moreover, although it is necessary to tell a screen transition engine about from which control (GUI components which generated the event) it was called, this is identified by passing Control ID (Ctrl ID) by the argument. That is, the screen which changes next from on a table using the value of this Ctrl ID is searched. as the notes in the case of using this function -- this function -- call appearance -- assignment of the transition place sometimes described in the table the bottom -- call appearance -- the dialog itself can be canceled the bottom

[0042] The procedure of the screen transition including cancellation of this dialog is explained concretely, referring to drawing 5 . Drawing 5 is drawing showing the procedure of the screen transition in the multi-window system of the information processor of drawing 1 .

[0043] Generating of the event to a user's input performs the call of a function

which makes a screen change from the GUI components which generated the event, as shown in drawing 5 . Control ID (Ctrl ID) is used for this call, and that argument is passed to a screen transition engine. The screen which changes next from on a table using the value of this Ctrl ID is searched with a screen transition engine. In canceling the dialog itself with this function, it performs the check of cancellation using a FLOW_DESTROY message to this dialog. In this case, a FLOW_CHILDDestroy message is used for a parent dialog and cancellation of a child dialog is notified to it. Here, the CALLBACKDATA structure is used when stopping cancellation. About this CALLBACKDATA structure, it mentions later.

[0044] On the other hand, if the dialog of a transition place is found out, screen transition to a transition place will be performed. In screen transition, it is displayed that it is exchanged so that a dialog may be created and the created dialog may lap.

[0045] Next, it explains, referring to drawing 6 and drawing 7 about the case where nesting of a dialog is performed using a modal dialog. Drawing and drawing 7 which show the screen transition by the nesting flow usual in drawing 6 are drawing showing the screen transition by the nesting flow in the

multi-window system of the information processor of drawing 1 .

[0046] In performing the nesting flow of a dialog using a modal dialog, as shown in drawing 6 , it takes the procedure of creating a modal dialog and usually creating a modal dialog once again in this dialog procedure.

[0047] In this usual procedure, when the transition to a dialog 3 arises from a dialog 1, it is necessary to create a dialog in a dialog 2, for example. Therefore, the code of creating a dialog 3 in the dialog procedure of a dialog 2 in this case must be described. On the other hand, with the gestalt of this operation, as shown in drawing 7 , all the dialogs that carry out nesting are created in a modeless dialog, from the dialog 1, it is controllable and all dialogs are made. Therefore, the nesting flow of a dialog can be realized, without making hard coding which was mentioned above. Moreover, in the example shown in drawing 7 , during the display period of a dialog 3, since a dialog 2 must be made into an invalid, the function which makes this dialog an invalid has been realized in a screen transition engine.

[0048] Here, the procedure of concrete processing of the screen transition to a user's event is explained, referring to drawing 8 and drawing 9 . The flow chart which shows the procedure of the processing about generation of a dialog [in /

in drawing 8 / the multi-window system of the information processor of drawing 1] and cancellation, the flow chart which shows the procedure in the interior of the function with which drawing 9 makes a screen change, and drawing 10 are drawings showing the configuration of the CALLBACKDATA structure. In addition, in this explanation, the condition that the screen transition engine was started and the dialog was displayed is assumed.

[0049] With reference to drawing 8 , when it is a message according [waiting and this inputted event] the input of an event to the carbon button depression on a dialog in step S141 first, it progresses to step S143, and read-out of the dialog information stored in memory 5 and writing are performed, and the screen transition function for making a screen change is called at continuing step S144. It shifts to the processing inside this function by call of this screen transition function (shown in drawing 9).

[0050] Inside this called screen transition function, as shown in drawing 9 , the dialog which should be first displayed on a degree in step S151 with reference to the transition place table TA 2 and the layered structure table TA 1 which are contained in screen transition table 1d is searched. At continuing step S152, the dialog group by which it is indicated by current with the dialog group displayed

on the degree obtained by retrieval with reference to the layered structure table TA 1 is compared, and the dialog which should be canceled, and the dialog which should be generated are listed. And it judges whether the dialog which should be progressed and canceled exists in step S153. When it judges whether the dialog which should be progressed and generated exists in step S154 when the dialog which should be canceled does not exist and the dialog which should be generated exists it, the dialog which should progress to step S155 and should be generated is generated, and it returns to the above-mentioned step S154 again. On the other hand, this processing is ended when the dialog which should be generated does not exist.

[0051] When judged with the dialog which should be canceled in step S153 existing, it progresses to step S158 and cancellation is checked to the dialog canceled first. And processing shifts to the event handler about the notice of dialog cancellation shown in drawing 8 (the above-mentioned step S141). If the notice of cancellation is checked in step S141 shown in drawing 8 , it will progress to step S145 and read-out of the dialog information stored in memory 5 and writing will be performed. This processing is processing performed if needed. Subsequently, it progresses to step S146 and judges whether based on the

internal information which a dialog has, a dialog is actually canceled. Here, when actually canceling a dialog, it cancels and returns to the above-mentioned step S141. On the other hand, if it judges with stopping cancellation of a dialog by a certain reason, it will progress to step S147 and the value which urges a cancellation termination to the cancellation termination flag `IntrptFlag` in the `CALLBACKDATA` structure (shown in drawing 10) transmitted from the screen transition engine will be set on the occasion of the notice of cancellation. This `CALLBACKDATA` structure is the convenience of application, and when stopping canceling for transition of a current dialog, it is the structure for notifying a cancellation termination to a screen transition engine. For example, if it usually becomes in case transition to `IDD_DLG3` from `IDD_DLG5` is performed as shown in drawing 18 , the dialog of `IDD_DLG5`, `IDD_DLG4`, and `IDD_DLG2` will be canceled in order. However, cancellation after it can be stopped by setting the above-mentioned cancellation termination flag `IntrptFlag` in the `CALLBACKDATA` structure, just before each dialog is canceled. Moreover, for example, if cancellation is stopped in the phase of `IDD_DLG4`, cancellation of each dialog of `IDD_DLG4` and `IDD_DLG2` will not be performed, but it will become the event waiting of the dialog screen of `IDD_DLG4` henceforth. .

[0052] By using this device, according to the conditions which can process arbitration in case screen transition is performed (namely, processing of step S145), and application holds, cancellation of a dialog can be checked (step S146) and cancellation can also be stopped (step S147). Moreover, inside a screen transition function, if it judges whether the above-mentioned cancellation termination flag IntrptFlg in the CALLBACKDATA structure is set in step S159 and the above-mentioned cancellation termination flag IntrptFlg is not set as shown in drawing 9 , it progresses to step S160, a dialog is canceled, and it returns to the above-mentioned step S153. It shifts to step S141 shown in drawing 8 with cancellation of this dialog, and judges that there was a notice of dialog cancellation in this step. And it progresses to step S142 and read-out of the dialog information stored in memory 5 and writing are performed. This processing is processing performed if needed.

[0053] Next, it explains, referring to drawing 11 thru/or drawing 15 about the approach of creating screen transition table 1d. Drawing 11 thru/or drawing 15 are drawings for explaining the creation approach of the screen transition table used for the multi-window system of the information processor of drawing 1 . In addition, the case where the layered structure of a dialog shown in drawing 42 is

table-ized is explained to an example here.

[0054] With the gestalt of this operation, in order to automate screen transition, two tables become important. That is, two tables with the transition place table TA 2 which can define a transition place as the layered structure table TA 1 which can express the layered structure of each dialog for every dialog are used to each dialog as shown in drawing 11 . In order to actually perform table-ization to each dialog, as shown in drawing 12 , concrete conditioning is added to each dialog. Specifically to each dialog, the resource ID (identifier) which starts in IDD_DLG in the dialog is assigned. For example, IDD_DLG1 is assigned to a dialog 1. Moreover, the dialog procedure DlgProc (program) which defines actuation is bound to each dialog. the identification number of a proper assigns to each dialog -- having -- **** -- this identification number -- the inside of one layered structure -- a dialog -- that sequence will not be asked if unique. However, assigning the number of "0" is not allowed. This is because it is used as an identification number for the number of "0" to specify transition termination, as mentioned later.

[0055] The relation between the dialogs in the application of the dialog base is possession relation fundamentally. The layered structure in such a configuration

is expressed by the table as shown in the next table 1.

[0056]

[Table 1]

ダイアログの識別 番号	ダイアログリソ ース I D	親ダイアログ の番号	ダイアログごとの遷 移先情報	グループ識別番号
1	IDD_DLG1	0	遷移先テーブル 1	1
2	IDD_DLG2	1	遷移先テーブル 2	1
3	IDD_DLG3	1	遷移先テーブル 3	1
4	IDD_DLG4	2	遷移先テーブル 4	1
5	IDD_DLG5	4	遷移先テーブル 5	1

Here, as a required item for expressing a layered structure in the above-mentioned table 1, the identification number of the identification number a dialog's, Dialog ID, and parents' dialog (owner window), the transition place information for every dialog, and the group number (refer to after-mentioned) are contained, and this transition place information is the start address (refer to after-mentioned) of a transition place table which defined which dialog is displayed by the inputted event. The table expressed in this table 1 is expressed using the construct of each programming language in a mounting phase.

[0057] When starting screen transition using this layered structure table, the data of a dialog which a top array element is made to start at the very beginning are described.

[0058] Next, the table definition of the transition place to the inputted event described to be "the transition place table 1" in Table 1 is explained.

[0059] Here, as shown in drawing 13 , three carbon button IDC_SET1, IDC_SET2, and IDC_CANCEL are arranged by the dialog of the identification number 1 shown in drawing 12 , and suppose at it that ID of a proper is assigned to each carbon button, respectively. ID of this carbon button is the identifier of a window resource, and this value is used for discernment of which carbon button the user chose. That is, to the dialog of the identification number 1 shown in drawing 12 , as shown in drawing 13 , when IDC_SET1 is chosen, when IDC_SET2 is chosen, it will change to the dialog (shown in drawing 15) of an identification number 3 to the dialog (shown in drawing 14) of an identification number 2, respectively, and the conditions of ending this transition when IDC_CANCEL is chosen will be set up. The example of description of the table for defining this transition place is shown in Table 2.

[0060]

[Table 2]

In addition, when the dialog identification number (ID) of a transition place is set to "0", the current line interrupted the screen transition control to require, and canceled all the dialogs by which it is indicated by current, and it has set up so that screen transition may be ended. Moreover, when two or more dialogs lap and are displayed at the time of termination, it cancels sequentially from the dialog arranged in the front face. The item in the above-mentioned table 1 "the transition place information for every dialog" stores the start address (if it says by C, it will be the pointer of the structure of a transition place table) of the table of this table 2, and, thereby, the carbon button and transition place on each dialog are associated.

[0061] Moreover, the dialog procedure (program) which defines actuation of Dialog ID and a dialog is associated on the table shown in Table 3, and each item of the start address of Dialog ID and a dialog procedure is included in this table.

[0062]

[Table 3]

Next, it explains, referring to drawing 16 thru/or drawing 19 about a hyperlink function and a sequential flow. Drawing for drawing 16 thru/or drawing 18 to explain the hyperlink function in the multi-window system of the information processor of drawing 1 and drawing 19 are drawings for explaining the sequential flow in the multi-window system of the information processor of drawing 1.

[0063] In the dialog which has the layered structure shown in above-mentioned drawing 11, supposing it is operational in the dialog of the present identification number 5, a screen is in the condition of having overlapped as shown in drawing 16. Here, by the conventional approach, one dialog is canceled at a time by the user in the condition which shows in drawing 16. On the other hand, with the gestalt of this operation, by defining windows other than an owner window in the transition place table of the dialog of an identification number 5 etc., it can jump directly in the dialog of an identification number 1 or an identification number 3

etc., and it can be canceled on the way. Thus, the function which other screens are made to change at a stretch, and is canceled on the way is called hyperlink function, and the example of screen transition by this hyperlink function is shown in drawing 17 and drawing 18 . What is necessary is just to describe the identification number of a dialog to display on the transition place table TA 2 at a degree, in order to specify such a transition place.

[0064] Moreover, it is possible to perform a sequential flow using a hyperlink function. For example, as shown in drawing 19 , when a dialog 2 changes in the dialog of an identification number 3 by the event from an operational condition in the dialog of the layered structure shown in drawing 11 , it seems that the dialog of a forefront side interchanged. That is, it means that the sequential flow was performed.

[0065] Thus, the layered structure table which described the layered structure of the dialog group which changes with the gestalt of this operation so that it may lap (TA1 shown in drawing 9), It is based on screen transition table 1d containing the transition place table (shown in drawing 9) TA 2 which described correspondence relation with the dialog which serves as a transition place of the dialog of the event inputted through the input device 4, and the this inputted

event from a user. Since it controls to search the dialog of a transition place and to display the searched dialog on the display means 6 to perform screen transition between dialogs automatically, screen transition including a nesting flow and a sequential flow can be performed efficiently.

[0066] In such a multi-window system, when the same dialog is repeatedly concerned with transition within screen transition table 1d, the resource which the dialog needs is shared and only two or more the transition places are defined. That is, dialog data are share-ized.

[0067] It explains referring to drawing 20 and drawing 21 about share-ization of this dialog data. Drawing and drawing 21 which show the example of a share of dialog data [in / in drawing 20 / the multi-window system of the information processor of drawing 1] are drawing showing the view of the screen transition in sharing of a dialog.

[0068] Here, as shown in drawing 20 , the case where the dialog of an identification number 3 is carrying out the dialog of an identification number 2 and the dialog of an identification number 1 as an owner window is considered. That is, as screen transition to which the dialog of an identification number 3 will be in an operational condition, there are two kinds of cases of the case where

the dialog of an identification number 3 laps with the dialog of an identification number 1, and the case where the dialog of an identification number 3 laps with the dialog of an identification number 2. And the transition is defined by the transition place table of the dialog of identification numbers 2 and 3, and in carrying out a nesting flow to the dialog of an identification number 1 to the identification number 2 to the dialog of an identification number 1 to the identification number 3 again, it does not produce a problem at all. However, when an event is inputted from the exterior to the dialog of an identification number 3, there is a case where he wants to change the screen displayed on a degree by through what kind of screen transition it passed.

[0069] If it explains concretely, the dialog of an identification number 3 will presuppose now that there is a Cancel button on a dialog in the operational condition. A current setup is canceled as a function of this Cancel button, and a dialog is canceled. However, by the description approach of an above-mentioned screen transition table, since only one transition place can be set up to one event, when it changes from the dialog of an identification number 2 to the dialog of an identification number 3, when it changes from the dialog of an identification number 1 to the dialog of an identification number 3, transition

which is said to the dialog of an identification number 1 cannot be described in one transition place table to the dialog of an identification number 2.

[0070] Then, the above-mentioned situation is avoided by considering the layered structure of a dialog, as shown in drawing 21 . Now, the dialog of an identification number 3 is separately considered to the thing and identification number 2 to the dialog of an identification number 1, respectively, and, specifically, each is set to a layered structure table. Thereby, a layered structure table can be described, as shown in Table 4.

[0071]

[Table 4]

That is, with the gestalt of this operation, the dialog of an identification number 3 is virtually interpreted as a separate thing on a layered structure table, and assignment of the transition place table is respectively enabled on the transition place table 3 and the transition place table 4. Moreover, since identifier

IDC_DL3 showing a window resource is using the same thing, an application code can completely use the same thing to the dialog 3 from which design data and the information on screen transition were separated, respectively.

[0072] Thus, RISOSU for realizing screen transition can be made into necessary minimum, as a result the burden to memory, such as an increment in memory space, can be made to mitigate by share-izing dialog data.

[0073] Moreover, in the gestalt of this operation, it is possible to carry out grouping of the transition of two or more dialogs to one collectively, and to describe per group. It explains referring to drawing 22 thru/or drawing 26 about this grouping. Drawing showing the actuation to the carbon button depression in a group when the layered structure Fig. showing the example which shares two or more dialogs which can set drawing 22 to the multi-window system of the information processor of drawing 1 , drawing showing the layered structure of the example to which drawing 23 carried out grouping of the dialog of drawing 22 , drawing showing the layered structure of the example to which drawing 24 carried out grouping of the 12 or less dialog of drawing 22 , and drawing 25 carry out grouping of two or more dialogs, and drawing 26 are drawings showing a group's example of cancellation.

[0074] In screen transition of the application of the dialog base, one function is set up on two or more screens in many cases. For example, as shown in drawing 22 , the 12 or less-dialog layered structure is share-ized. However, by this example, subsequent screen transition differs from share-ization of a dialog having described similarly by the case where it laps with the front face of the case where the dialog 12 currently shared laps with the front face of a dialog 5, and a dialog 11. Then, it is considered like the approach of carrying out share-ization of dialog data that 12 or less dialog is virtually separate. It enables this to express the share on a layered structure using the same window resource. Furthermore, the device which carries out grouping of this dialog group to one collectively is introduced. I hear that the merit which carries out grouping of the dialog group collectively can change into arbitration the transition place of the dialog which belongs in a group, and there is. That is, it becomes possible to set up the transition place according to an individual by the difference in transition from the dialog 11 shown, for example in drawing 22 .

[0075] When these things are taken into consideration, the layered structure shown in drawing 22 can be rewritten to the layered structure shown in drawing 23 and drawing 24 , respectively, and what table-ized each layered structure,

respectively is shown in Table 5 and Table 6.

[0076]

[Table 5]

[0077]

[Table 6]

In the table shown in Table 5, a group's 2 dialog group is assigned by the dialog

identification numbers 12 and 13 into the group 1. Moreover, when it changes from the dialog of either of the groups 1 to a group 2, i.e., the dialog of an identification number 12 or an identification number 13, it is set up so that the dialog described at the very head of the tables which express a group's 2 layered structure first may start. Here, although grouping of the case of an ordinary dialog was carried out to the item of a group identification number in the number of the group to whom the dialog belongs, the group's number is specified as a case.

[0078] Although it may be good as long as the identification number of the dialog described in a group's 2 layered structure is unique within a group, and the number currently used in the group 1 may be overlapped, since it is necessary to carry out unique [of the transition place table] with application, making the structure name currently used within the group 1 overlap must avoid it.

[0079] In addition, one table is created for registration of a group's identification number, and a group's identification number currently used here uses the number defined within the group table of Table 7. The item of this group table is the order of a group's identification number and the start address of a layered structure table.

[0080]

[Table 7]

Next, explanation of the format of the transition place table at the time of considering grouping is calling the group 2 from the group 1 in the example shown in drawing 22 . And when transition shifts to a group 2 from a group 1, screen transition is performed according to description of the transition place table defined to each dialog of a group 2. However, transition of the dialog in a group 2 is customized in many cases depending on the transition from which dialog in a group 1 it is.

[0081] In such a case, it receives and then customizes like. When it presupposes that there is a carbon button on the dialog 2-2 in a group 2 in the layered structure of a dialog shown in drawing 25 and a carbon button is pushed on the transition place table of this dialog 2-2, it shall be described that it changes in the dialog 2-1 in a group 2. Here, it considers customizing a transition place when the carbon button on the dialog 2-2 in a group 2 is pushed by the case where it

changes in a group 2 as the case where saw from the group 1 side and it changes in a group 2 as an identification number 12, and an identification number 13. When ID of the carbon button on the dialog 2-2 in this group 2 is made into IDC_BTN1 and this carbon button IDC_BTN1 is chosen, it is set up so that it may change to a group's 2 dialog 2-1 by the default. And when having changed as a group 2 of an identification number 12 within a group 1, and having changed the transition place to the depression of carbon button IDC_BTN1 as a group 2 of an identification number 13 in the dialog 5, the example which customized the transition place to the depression of carbon button IDC_BTN1 so that it might become a dialog 11 is shown.

[0082] First, the transition place table to the dialog 2-2 in the group 2 in the layered structure of a dialog shown in drawing 25 which defines the destination of a dialog is expressed to Table 8. This table turns into a default transition place table.

[0083]

[Table 8]

Therefore, when you make it connected with the transition in a group 1 and carbon button IDC_BTN1 of a group's 2 dialog 2-2 is not customized, transition actuation is performed according to this transition place table. And in order to change so that it may change to the dialog 5 in a group 1 when carbon button IDC_BTN1 is chosen on the dialog 2-2 in a group 2, it is described that the group 2 (the identification number 12 is assigned) who makes the dialog 5 in a group 1 the owner expresses the destination customized on the transition place table of a group's 1 identification number 12 to Table 9.

[0084]

[Table 9]

Here, when this description does not exist, the transition place defined within the default transition place table (transition place table 2-2) of the dialog 2-2 in a group 2 is referred to. Thus, if the transition place is customized, the direction of the customized transition place will be referred to preferentially. Moreover,

customize of a transition place is possible to every control ID in the group who changes. For this reason, a dialog implementer needs to take care that ID assigned to the control by which a transition place may be customized does not overlap within the same group. In addition, even if it is not necessary to customize, it is described that a transition place table has the element of "0" individuals as it must create and is shown in Table 10. The transition place table which similarly defines the transition place to the group 2 currently assigned to the dialog number 13 in a group 1 is described as shown in Table 11.

[0085]

[Table 10]

[0086]

[Table 11]

Furthermore, since there are some notes on the occasion of grouping and

share-izing, it is explained.

[0087] When carbon button IDC_BTN1 of a group's 2 dialog 2-2 is pushed in the layered structure of a dialog shown in above-mentioned drawing 25 , the case where it is specified that it ends transition is considered. That is, the case where "0" is specified as the identification number of the dialog of a destination on a transition place table is considered. Usually, when transition is not continuing among two or more groups, all the dialogs by which it is indicated by current are canceled, and it ends. On the other hand, as shown in drawing 26 , when hierarchical relationship is between a group 1 and a group 2 and a termination instruction appears in the transition in a group 2, only the dialog belonging to the group is canceled. Although the group 2 is defined as a dialog identification number 12 in a group 1 when canceling all the dialogs belonging to the group 1 of the side called temporarily here and making it want to end What is necessary is to customize the transition place of ID of the carbon button on the dialog in a group 2 in the transition place table which defines this transition place, and just to specify termination (namely, a destination "0") of transition within a group 1. It becomes as it is shown in actual description, then Table 12 about this.

[0088]

[Table 12]

Next, it explains, referring to drawing 27 and drawing 28 about the message notification device at the time of canceling a dialog. Drawing in which drawing 27 shows an example of cancellation of a dialog, and drawing 28 are drawings showing the procedure of a message notification device.

[0089] Here, as shown in drawing 27 , screen transition when the dialog of an identification number 2 is canceled by the depression of the carbon button on the dialog of an identification number 2 and only the dialog of an identification number 1 remains is explained to an example.

[0090] In case a dialog is canceled, first, according to the demand of that check which may actually cancel the dialog which corresponds from a screen transition function, a FLOW_DESTROY message is transmitted from a screen transition engine to a dialog, and the post process of a dialog is performed to drawing 28 in the handler corresponding to this so that it may be shown. Moreover, it is set up by setting up the value of IntrptFlag from a dialog procedure whether that dialog

is canceled at the same time delivery of the information at the time of cancellation is performed using the CALLBACKDATA structure (shown in drawing 10) mentioned above on this occasion. IntrptFlag is set as TRUE to interrupt cancellation of a dialog according to some conditions temporarily. Moreover, ReturnID in the CALLBACKDATA structure is applicable to the judgment of whether to interrupt the cancellation, when each dialog is canceled by using either of the values which could specify as an argument of a screen transition function, for example, were defined [GO BACK / O.K., CANCEL,] beforehand. And after the cancellation is completed, a FLOW_CHILDDESTORY message is transmitted to the owner window of the dialog, and it can know that the window which self owns was canceled. Moreover, if required, a certain processing will be performed in the message handler. If this check is once transmitted to all dialogs by transition and interruption of cancellation is checked by that middle, cancellation will be interrupted at that time and transition from that screen will be continued.

[0091] With the gestalt of this operation, the specific dialog which disregarded screen transition table 1d and the user set up is displayed, and this dialog can be canceled, without performing screen transition to a user's event inputted on this

specific dialog. It explains referring to drawing 29 about this device. Drawing 29 is drawing showing the example of a display of the dialog in the multi-window system of the information processor of drawing 1 set up dynamically.

[0092] The dialog located in a deep hierarchy as one of the functions of the application of the applied dialog base is registered, and there is a function to call only the dialog. However, since transition will be fundamentally realized with reference to screen transition table 1d once it starts screen transition, it is difficult to display a dialog based on the dynamic screen transition table registered as the user was screen transition. Then, in order to realize the function, the layered structure of a window is disregarded temporarily, and it constitutes so that a specific screen can be displayed.

[0093] The merit of improving the reusability of a dialog procedure is obtained by this function. Since the usual dialog procedure makes screen transition perform, it is the code which can call a screen transition function in the event handler to all button grabbing. However, now, the dialog procedure usually used cannot be used as it is but a thing other than what is usually being used must be created to perform actuation which closes a dialog, without displaying one of dialogs and making it change anywhere. Then, the function which disregards the layered

structure newly described by screen transition table 1d in the specific dialog, and is displayed on a forefront side is prepared so that the dialog procedure usually used can be used as it is. This function is for displaying the dialog specified as the forefront side of the screen which has changed on the thread which called that function. That is, the above-mentioned function makes the screen cancel without making screen transition perform, even if a screen transition function is called in the dialog procedure of the dialog currently displayed. Therefore, the dialog procedure usually used can use as it is, and leads to improvement in development effectiveness, and saving of memory. For example, as shown in drawing 29 , this function can continue the condition of the screen which had changed before as it is, when temporary dialog DLG_TMP is displayed and that dialog is canceled after that.

[0094] Thread ID is used as a key and the information which is needed when performing screen transition manages it so that screen transition can be realized independently on each thread on OS which supported the multithread. It explains referring to drawing 30 thru/or drawing 36 about this management method. Drawing, drawing 33 , or drawing 36 which shows the example of screen transition for drawing for drawing 30 to show the hierarchical relationship

of a dialog and the maintenance approach of internal information in the multi window of the information processor of drawing 1 , drawing showing the structure to which drawing 31 holds the internal information of drawing 30 , and drawing 32 to explain transition of the internal information of drawing 30 is drawing showing transition of internal information.

[0095] In an above-mentioned management method, as shown in drawing 30 , the layered structure of a dialog is transposed to a stack-like information structure, and is held. It is canceled, after the FLOWMEMMGR structure is created corresponding to each flow and a flow is specifically completed, as shown in drawing 31 . Moreover, this structure has the FLOWSTACK structure in each, and he is trying to manage the nest situation of a dialog, and the information on each dialog proper in the shape of a stack by making the information with which the FLOWSTACK structure was loaded correspond with the layered structure of a dialog.

[0096] Here, the semantics of the member of the FLOWMEMMGR structure and the FLOWSTACK structure shown in drawing 31 is as follows.

FLOWMEMMGR:DWORD The pointer HWND RootWnd of the pointer FLOWSTACK *FlowStack;FLOWSTACK array of structures (stack) to the

structure in which ThreadID; thread ID TREETABLE *TreeTable; Tree information was stored; a flow The handle WORD StackNum of the started window; Flag FLOWSTACK:WORD in case counter BOOL fCallDlg; CallDialog showing with how many data the current stack is loaded is used DlgNum; dialog identification number

WORD DlgID; dialog resource ID HWND hWnd; -- window-handle HWND hWndParent; of a dialog -- parents' window-handle BOOL Check; -- the group number of the table under flag WORD TreeNum; current transition which confirms whether a window is canceled or not

WORD GroupNum; the group number within the table of a branching agency

Pointer TRRETTABLE *PrevTable of the TREETTABLE structure corresponding to a TREETTABLE *CurrentTable; dialog identification number; the processing in the interior of the screen transition function performed with generating of actual screen transition by the pointer of the TREETTABLE structure corresponding to the group number within a branching agency group's table is explained. Here, as shown in drawing 32 , the case where a screen changes from a dialog 4 to a dialog 3 is shown in an example. In addition, the example shown below corresponds to the flow chart of already explained drawing 8 and

drawing 9 .

[0097] First, the function which starts screen transition is started from the dialog of flow initiation. When another flow is started while developing one flow since only one flow can be developed to one thread as mentioned above, the already started flow shall be ended and a new flow shall be started. In addition, a value is set to one of the FLOWMEMMGR structures while setting Thread ID to a key, if a flow is started. And one dialog starts, the FLOWSTACK structure is also created to the structure, and the first dialog information is stored.

[0098] When the first dialog is created, the function which makes a screen change will be called from a degree. The following processings are performed inside this function.

[0099] That is, if the in vent from a user is inputted (step S141), the thread ID of the dialog in which actuation is performed now will be acquired, and the stack information about the dialog which operated the carbon button for the thread ID to the key will be acquired. Based on this acquired information, the dialog displayed next on a screen from a layered structure table and a transition place table is searched (step S151). However, when "0" is specified as the dialog number of the destination to control specified as the transition place table, it

returns to the dialog which canceled all the dialogs and started screen transition (step S141).

[0100] Another stack for expressing the hierarchical relationship is built, following the hierarchical relationship to the dialog, as it is shown in drawing 33 , when it determines which dialog is displayed on a screen (step S152).

[0101] Subsequently, what was created by another stack is compared with the present stack, and when there is a common dialog in which both of the stacks exist, TRUE is set to the Check member of the structure corresponding to the dialog of the present stack as shown in drawing 34 (step S152).

[0102] And since the dialog to which Check is not attached in the present stack must cancel, these are canceled (step S153, step S160). However, when the flag (step S158, step S159) which interrupts cancellation stands since a check is taken in the dialog in case it cancels as are shown in drawing 35 , and mentioned above, screen transition is ended there and it waits for the input of an event -- the following carbon button is pushed -- (step S141).

[0103] Except the dialog common at this time, when there is no dialog which should be canceled (step S153), as shown in drawing 36 , since it should be canceled, every one dialog after a common dialog is created and piled up from

the stack newly created next (step S155), and that information is pushed to coincidence at the present stack.

[0104] And it shifts to the processing which calls the function which makes a screen change from this created dialog again (step S141).

[0105]

[Effect of the Invention] As explained above, according to the multi-window system according to claim 1 The table which described the layered structure of the dialog group which changes so that it may lap, The screen transition table containing the table which described correspondence relation with the dialog used as the transition place of the dialog of the event inputted by the user and the this inputted event, Since it has the screen transition control means controlled to perform screen transition between dialogs based on a screen transition table, screen transition including a nesting flow and a sequential flow can be performed efficiently.

[0106] According to the multi-window system according to claim 2, since the screen transition table is able to define two or more transition places to a modal dialog which carries out a multiple-times appearance using the same resource, necessary minimum of the resource for realizing screen transition can be carried

out, and the volume load to storage means, such as memory, can be mitigated as a result.

[0107] Since a screen transition table can carry out grouping of the transition of two or more dialogs to one collectively and can describe per group, in case a screen transition table is described according to the multi-window system according to claim 3, the time and effort which overlaps and performs the same description can be saved, and the time and effort of excessive description can be saved, a description mistake can be reduced, and the screen transition table excellent in readability can be created.

[0108] According to the multi-window system according to claim 4, since a screen transition table can change the transition place to the event of the user of the dialog by the side of [which was shared] it to the dialog and the dialog group to whom share-ization was given according to the transition situation till then, it becomes possible to carry out minimization of RISOSU which is originally the relation of a trade-off, the increase in efficiency of screen transition table description, and a setup of a fine transition place in coexistence.

[0109] According to the multi-window system according to claim 5, further, when a dialog is canceled according to a screen transition table, since it has a

cancellation check means to check cancellation of this dialog, and a notice means to notify cancellation of this dialog to the owner window of this dialog when cancellation of a dialog is checked by the cancellation check means, in case screen transition is performed, the status currently held in connection with it can be changed. Moreover, it becomes possible to check cancellation of a dialog and to stop cancellation according to the conditions which application holds.

[0110] According to the multi-window system according to claim 6, the specific dialog which disregarded the screen transition table and the user set up further is displayed. Since it has a cancellation means to cancel this dialog, without performing screen transition to a user's event inputted on this specific dialog It becomes possible not only the screen transition that used only the static data described on the screen transition table but to change screens which change dynamically, such as screen transition to the dialog registered while the user operated the screen.

[0111] According to the screen transition approach and a storage according to claim 13 according to claim 7, screen transition including a nesting flow and a sequential flow can be performed efficiently.

[0112] According to the screen transition approach and a storage according to

claim 14 according to claim 8, necessary minimum of the resource for realizing screen transition can be carried out, and the volume load to storage means, such as memory, can be mitigated as a result.

[0113] In case a screen transition table is described according to the screen transition approach and a storage according to claim 15 according to claim 9, the time and effort which overlaps and performs the same description can be saved, and the time and effort of excessive description can be saved, a description mistake can be reduced, and the screen transition table excellent in readability can be created.

[0114] According to the screen transition approach and a storage according to claim 16 according to claim 10, it becomes possible to carry out minimization of RISOSU which is originally the relation of a trade-off, the increase in efficiency of screen transition table description, and a setup of a fine transition place in coexistence.

[0115] According to the screen transition approach and a storage according to claim 17 according to claim 11, in case screen transition is performed, the status currently held in connection with it can be changed. Moreover, it becomes possible to check cancellation of a dialog and to stop cancellation according to

the conditions which application holds.

[0116] According to the screen transition approach and a storage according to claim 18 according to claim 12, it becomes possible not only the screen transition that used only the static data described on the screen transition table but to change screens which change dynamically, such as screen transition to the dialog registered while the user operated the screen.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the configuration of the information processor for building the 1st gestalt of operation of the multi-window system of this invention.

[Drawing 2] It is the conceptual diagram showing the configuration for realizing screen transition control in the multi window currently built on the information processor of drawing 1 .

[Drawing 3] It is the flow chart which shows the procedure of the screen transition in the multi-window system currently built on the information processor of drawing 1 .

[Drawing 4] It is drawing showing the procedure of the screen transition initiation in the multi-window system of the information processor of drawing 1 .

[Drawing 5] It is drawing showing the procedure of the screen transition in the multi-window system of the information processor of drawing 1 .

[Drawing 6] It is drawing showing the screen transition by the usual nesting flow.

[Drawing 7] It is drawing showing the screen transition by the nesting flow in the multi-window system of the information processor of drawing 1 .

[Drawing 8] It is the flow chart which shows the procedure of the processing about generation of the dialog in the multi-window system of the information processor of drawing 1 , and cancellation.

[Drawing 9] It is the flow chart which shows the procedure in the interior of the function which makes a screen change.

[Drawing 10] It is drawing showing the configuration of the CALLBACKDATA structure.

[Drawing 11] It is drawing for explaining the creation approach of the screen transition table used for the multi-window system of the information processor of drawing 1 .

[Drawing 12] It is drawing for explaining the creation approach of the screen transition table used for the multi-window system of the information processor of drawing 1 .

[Drawing 13] It is drawing for explaining the creation approach of the screen transition table used for the multi-window system of the information processor of drawing 1 .

[Drawing 14] It is drawing for explaining the creation approach of the screen transition table used for the multi-window system of the information processor of drawing 1 .

[Drawing 15] It is drawing for explaining the creation approach of the screen transition table used for the multi-window system of the information processor of drawing 1 .

[Drawing 16] It is drawing for explaining the hyperlink function in the multi-window system of the information processor of drawing 1 .

[Drawing 17] It is drawing for explaining the hyperlink function in the multi-window system of the information processor of drawing 1 .

[Drawing 18] It is drawing for explaining the hyperlink function in the multi-window system of the information processor of drawing 1 .

[Drawing 19] It is drawing for explaining the sequential flow in the multi-window system of the information processor of drawing 1 .

[Drawing 20] It is drawing showing the example of a share of the dialog data in the multi-window system of the information processor of drawing 1 .

[Drawing 21] It is drawing showing the view of the screen transition in sharing of a dialog.

[Drawing 22] It is the layered structure Fig. showing the example which shares two or more dialogs which can be set to the multi-window system of the information processor of drawing 1 .

[Drawing 23] It is drawing showing the layered structure of the example which carried out grouping of the dialog of drawing 22 .

[Drawing 24] It is drawing showing the layered structure of the example which carried out grouping of the 12 or less dialog of drawing 22 .

[Drawing 25] It is drawing showing the actuation to the carbon button depression in the group at the time of carrying out grouping of two or more dialogs.

[Drawing 26] It is drawing showing a group's example of cancellation.

[Drawing 27] It is drawing showing an example of cancellation of a dialog.

[Drawing 28] It is drawing showing the procedure of a message notification device.

[Drawing 29] It is drawing showing the example of a display of the dialog in the multi-window system of the information processor of drawing 1 set up dynamically.

[Drawing 30] It is drawing to show the hierarchical relationship of a dialog and the maintenance approach of internal information in the multi window of the

information processor of drawing 1 .

[Drawing 31] It is drawing showing the structure holding the internal information of drawing 30 .

[Drawing 32] It is drawing showing the example of screen transition for explaining transition of the internal information of drawing 30 .

[Drawing 33] It is drawing showing transition of internal information.

[Drawing 34] It is drawing showing transition of internal information.

[Drawing 35] It is drawing showing transition of internal information.

[Drawing 36] It is drawing showing transition of internal information.

[Drawing 37] It is drawing showing the example of screen transition by the nesting flow which is one of the screen transition approaches of a multi-window system.

[Drawing 38] It is drawing showing the tree structure of the window in the case of a nesting flow.

[Drawing 39] It is drawing showing the example of screen transition by the sequential flow which is other one of the screen transition approaches of a multi-window system.

[Drawing 40] It is drawing showing the tree structure of the window in the case of

generating the screen transition by the sequential flow.

[Drawing 41] A dialog is drawing showing the only extinguished example of screen transition.

[Drawing 42] It is drawing showing an example showing screen transition of a window layered structure.

[Description of Notations]

1 Multi-window Program

1a Screen information attaching part

1b Screen transition control section

1c Screen-display section

1d Screen transition table

2 OS

3 CPU

4 Input Device

5 Memory

6 Display Means

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-105654

(P2000-105654A)

(43) 公開日 平成12年4月11日 (2000.4.11)

(51) Int.Cl.⁷

G 0 6 F 3/00

識別記号

6 5 5

F I

G 0 6 F 3/00

テマコード* (参考)

6 5 5 A 5 E 5 0 1

審査請求 未請求 請求項の数18 F D (全 23 頁)

(21) 出願番号

特願平10-288679

(22) 出願日

平成10年9月28日 (1998.9.28)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 柴尾 弘毅

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(72) 発明者 渡辺 浩康

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 100081880

弁理士 渡部 敏彦

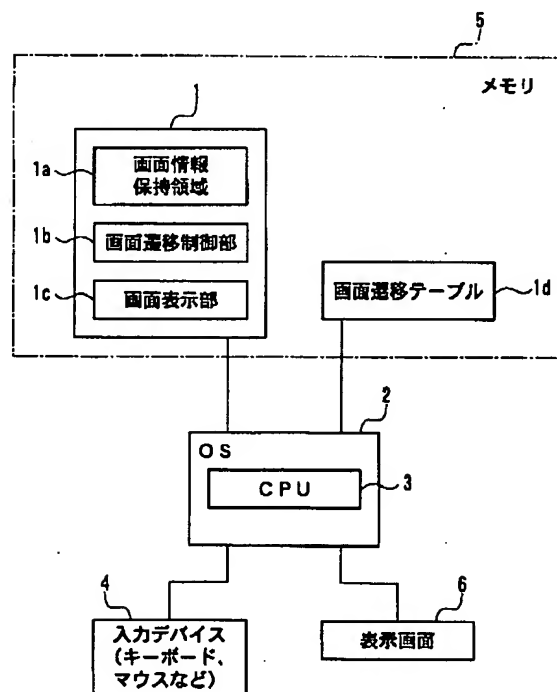
最終頁に続く

(54) 【発明の名称】 マルチウィンドウシステム、画面遷移方法および記憶媒体

(57) 【要約】

【課題】 ネスティングフロー、シーケンシャルフローを含む画面遷移を効率良く実行することができるマルチウィンドウシステムを提供する。

【解決手段】 マルチウィンドウプログラム1は、画面遷移テーブル1dを含む。画面遷移テーブル1dには、重なるように遷移するダイアログボックス群の階層構造を記述した階層構造テーブル（図9に示すTA1）と、ユーザから入力デバイス4を介して入力されたイベントと該入力されたイベントのダイアログボックスの遷移先となるダイアログボックスとの対応関係を記述した遷移先テーブル（図9に示す）TA2とが含まれている。画面遷移制御部1bは、画面遷移テーブル1dに基づきダイアログボックス間の画面遷移を自動的に行うように遷移先のダイアログボックスを検索し、画面表示部1cは検索されたダイアログボックスを表示手段6に表示するように制御する。



【特許請求の範囲】

【請求項1】 表示手段上に、ユーザとの対話形式の画面を表示するマルチウィンドウシステムにおいて、重なるように遷移するダイアログ群の階層構造を記述したテーブルと、ユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルとを含む画面遷移テーブルと、前記画面遷移テーブルに基づき前記ダイアログ間の画面遷移を行うように制御する画面遷移制御手段とを備えることを特徴とするマルチウィンドウシステム。

【請求項2】 前記画面遷移テーブルは、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であることを特徴とする請求項1記載のマルチウィンドウシステム。

【請求項3】 前記画面遷移テーブルは、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であることを特徴とする請求項1記載のマルチウィンドウシステム。

【請求項4】 前記画面遷移テーブルは、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であることを特徴とする請求項1記載のマルチウィンドウシステム。

【請求項5】 さらに、前記画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認する破棄確認手段と、前記破棄確認手段により前記ダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知する通知手段とを備えることを特徴とする請求項1記載のマルチウィンドウシステム。

【請求項6】 さらに、前記画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力された前記ユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄する破棄手段を備えることを特徴とする請求項1記載のマルチウィンドウシステム。

【請求項7】 表示手段上に、ユーザとの対話形式の画面を表示するマルチウィンドウシステムに用いられる画面遷移方法において、重なるように遷移するダイアログ群の階層構造を記述したテーブルおよびユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルを含む画面遷移テーブルを作成し、前記画面遷移テーブルに基づき前記ダイアログ間の画面遷移を行うように制御することを特徴とする画面遷移方法。

【請求項8】 前記画面遷移テーブルは、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であること

を特徴とする請求項7記載の画面遷移方法。

【請求項9】 前記画面遷移テーブルは、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であることを特徴とする請求項7記載の画面遷移方法。

【請求項10】 前記画面遷移テーブルは、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であることを特徴とする請求項7記載の画面遷移方法。

【請求項11】 さらに、前記画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認し、前記ダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知することを特徴とする請求項7記載の画面遷移方法。

【請求項12】 さらに、前記画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力された前記ユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄することを特徴とする請求項7記載の画面遷移方法。

【請求項13】 表示手段上に、ユーザとの対話形式の画面を表示するマルチウィンドウシステムを構築するためのプログラムを格納した記憶媒体において、前記プログラムは、重なるように遷移するダイアログ群の階層構造を記述したテーブルおよびユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルを含む画面遷移テーブルと、前記画面遷移テーブルに基づき前記ダイアログ間の画面遷移を行うように制御する画面遷移制御モジュールとを含むことを特徴とする記憶媒体。

【請求項14】 前記画面遷移テーブルは、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であることを特徴とする請求項13記載の記憶媒体。

【請求項15】 前記画面遷移テーブルは、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であることを特徴とする請求項13記載の記憶媒体。

【請求項16】 前記画面遷移テーブルは、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であることを特徴とする請求項13記載の記憶媒体。

【請求項17】 前記プログラムは、さらに、前記画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認する破棄確認モジュールと、前記破棄確認モジュールにより前記ダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知する通知モジュールとを含む

ことを特徴とする請求項13記載の記憶媒体。

【請求項18】 前記プログラムは、さらに、前記画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力された前記ユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄する破棄モジュールを含むことを特徴とする請求項13記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マルチウィンドウシステム、それに用いられる画面遷移方法および記憶媒体に関する。

【0002】

【従来の技術】複写機などのOA機器などにおいては、通常、所望するモードの設定指示、該モードに関する設定値などを入力するための操作部が設けられている。現在、この操作部としては、入力操作画面を液晶表示パネルなどの表示手段に表示し、この入力操作画面を用いて上記入力者対話形式で行うように構成されているものが主流になりつつある。この操作部に用いられる、入力操作画面を生成して表示するマルチウィンドウシステムは、複数のダイアログ間を遷移することにより入力操作画面を遷移させるように構成されているものが多く、ユーザにより操作部上でダイアログの生成、破棄を繰り返しながら設定が進められる。

【0003】このようなウィンドウシステムに用いられている画面遷移方法について図37ないし図42を参照しながら説明する。図37はマルチウィンドウシステムの画面遷移方法の一つであるネスティングフローによる画面遷移例を示す図、図38はネスティングフローの場合におけるウィンドウのトリート構造を示す図、図39はマルチウィンドウシステムの画面遷移方法の他の一つであるシーケンシャルフローによる画面遷移例を示す図、図40はシーケンシャルフローによる画面遷移が発生する場合におけるウィンドウのトリート構造を示す図、図41はダイアログが単に消滅する画面遷移例を示す図、図42は画面遷移を表すウィンドウ階層構造の一例を示す図である。

【0004】まず、ネスティングフローについて説明すると、図37に示すように、ネスティングフローは、ダイアログがユーザからのイベント（一般的にはダイアログ上のボタンの押下）によって、その上に重なるように遷移するものである。具体的には、ダイアログ1上でボタンが押下されると、この押下されたボタンに対応するダイアログ2がダイアログ1上に重ねて表示される。このような遷移が起こる場合においては、ウィンドウの所有関係が図38に示すトリート構造で表される。

【0005】これに対し、シーケンシャルフローは、図39に示すように、ダイアログがユーザからのイベントによって入れ替わるように遷移するものである。具体的

には、最前面に表示されているダイアログ1上でユーザからイベントが入力されると（ボタンが押下されると）、ダイアログ1が一旦破棄されて押下されたボタンに対応するダイアログ2が表示される。この遷移におけるウィンドウの所有関係は、図40に示すトリート構造で示され、ダイアログ1とダイアログ2とは同じ階層にある。

【0006】また、図41に示すように、表示されているダイアログが単に消滅するような遷移もある。

【0007】基本的にダイアログを用いたアプリケーションの画面遷移は、上記遷移の組合せによって表現することができ、そのウィンドウ階層構造は、例えば図38および図40に示す構造を組み合わせることによって得られ、その構造は図42に示すような構造からなる。通常、一つの画面には複数のボタンが表示され、ユーザが押下したボタンに応じて遷移する画面または動作が変わる。例えば、図42に示す階層構造では、ダイアログ1が表示されているときに「上」のボタンが押下されると、ダイアログ2が、「下」のボタンが押下されると、ダイアログ3がダイアログ1の前面に重なるように表示されることになる。

【0008】このような画面遷移動作は、アプリケーションのコード内でハードコーディングされていることが一般的であったが、アプリケーションコードの再利用やツールを利用した画面遷移制御を可能にするために、画面遷移の情報をアプリケーションから分離、テーブル化し、それを参照することによって画面遷移を制御する必要性が生じ、この画面遷移テーブルを用いて、ユーザからのイベントに応じて自動的に次々と画面を入れ替えることが可能にする技術が既に提案されている。この提案されている技術として、特開平6-168203号公報、特開平9-097158号公報などに記載されているものがある。具体的には、両者には、画面を順に入れ替えていく遷移の自動化技術が記載されている。

【0009】また、特開平8-016350号公報には、深い階層のダイアログを遷移させる手順を簡易化する画面遷移方法が示されている。

【0010】

【発明が解決しようとする課題】しかし、特開平6-168203号公報、特開平9-097158号公報に記載の技術では、画面を順に入れ替えていく遷移の自動化を行うものであって、ダイアログを前面に積み重ねるように表示する画面遷移を自動的に行うことはできない。すなわち、ダイアログを前面に積み重ねるように表示する画面遷移を自動的に行うための画面遷移テーブルの書式が確立されていない。

【0011】また、特開平8-016350号公報に記載の画面遷移方法には、上述したネスティングフローの概念がないとともに、ダイアログ間の所有関係が考慮されていないから、ネスティングフローとシーケンシャル

フローに関して、画面遷移テーブルを使用して直接の所有関係を持たないダイアログに自動的に遷移するような画面切替を行うことができない。

【0012】本発明の目的は、ネスティングフロー、シーケンシャルフローを含む画面遷移を効率良く実行することができるマルチウィンドウシステム、画面遷移方法および記憶媒体を提供することにある。

【0013】

【課題を解決するための手段】請求項1記載の発明は、表示手段上に、ユーザとの対話形式の画面を表示するマルチウィンドウシステムにおいて、重なるように遷移するダイアログ群の階層構造を記述したテーブルと、ユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルとを含む画面遷移テーブルと、前記画面遷移テーブルに基づき前記ダイアログ間の画面遷移を行うように制御する画面遷移制御手段とを備えることを特徴とする。

【0014】請求項2記載の発明は、請求項1記載のマルチウィンドウシステムにおいて、前記画面遷移テーブルは、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であることを特徴とする。

【0015】請求項3記載の発明は、請求項1記載のマルチウィンドウシステムにおいて、前記画面遷移テーブルは、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であることを特徴とする。

【0016】請求項4記載の発明は、請求項1記載のマルチウィンドウシステムにおいて、前記画面遷移テーブルは、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であることを特徴とする。

【0017】請求項5記載の発明は、請求項1記載のマルチウィンドウシステムにおいて、さらに、前記画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認する破棄確認手段と、前記破棄確認手段により前記ダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知する通知手段とを備えることを特徴とする。

【0018】請求項6記載の発明は、請求項1記載のマルチウィンドウシステムにおいて、さらに、前記画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力された前記ユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄する破棄手段を備えることを特徴とする。

【0019】請求項7記載の発明は、表示手段上に、ユーザとの対話形式の画面を表示するマルチウィンドウシステムに用いられる画面遷移方法において、重なるよう

に遷移するダイアログ群の階層構造を記述したテーブルおよびユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルを含む画面遷移テーブルを作成し、前記画面遷移テーブルに基づき前記ダイアログ間の画面遷移を行うように制御することを特徴とする。

【0020】請求項8記載の発明は、請求項7記載の画面遷移方法において、前記画面遷移テーブルは、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であることを特徴とする。

【0021】請求項9記載の発明は、請求項7記載の画面遷移方法において、前記画面遷移テーブルは、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であることを特徴とする。

【0022】請求項10記載の発明は、請求項7記載の画面遷移方法において、前記画面遷移テーブルは、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であることを特徴とする。

【0023】請求項11記載の発明は、請求項7記載の画面遷移方法において、さらに、前記画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認し、前記ダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知することを特徴とする。

【0024】請求項12記載の発明は、請求項7記載の画面遷移方法において、さらに、前記画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力された前記ユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄することを特徴とする。

【0025】請求項13記載の発明は、表示手段上に、ユーザとの対話形式の画面を表示するマルチウィンドウシステムを構築するためのプログラムを格納した記憶媒体において、前記プログラムは、重なるように遷移するダイアログ群の階層構造を記述したテーブルおよびユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルを含む画面遷移テーブルと、前記画面遷移テーブルに基づき前記ダイアログ間の画面遷移を行うように制御する画面遷移制御モジュールとを含むことを特徴とする。

【0026】請求項14記載の発明は、請求項13記載の記憶媒体において、前記画面遷移テーブルは、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であることを特徴とする。

【0027】請求項15記載の発明は、請求項13記載

の記憶媒体において、前記画面遷移テーブルは、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であることを特徴とする。

【0028】請求項16記載の発明は、請求項13記載の記憶媒体において、前記画面遷移テーブルは、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であることを特徴とする。

【0029】請求項17記載の発明は、請求項13記載の記憶媒体において、前記プログラムは、さらに、前記画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認する破棄確認モジュールと、前記破棄確認モジュールにより前記ダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知する通知モジュールとを含むことを特徴とする。

【0030】請求項18記載の発明は、請求項13記載の記憶媒体において、前記プログラムは、さらに、前記画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力された前記ユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄する破棄モジュールを含むことを特徴とする。

【0031】

【発明の実施の形態】以下に本発明の実施の形態について図を参照しながら説明する。

【0032】図1は本発明のマルチウィンドウシステムの実施の一形態を構築するための情報処理装置の構成を示すブロック図である。

【0033】情報処理装置は、図1に示すように、OS 2に従い装置全体の制御を行うとともに、メモリ5に格納されているマルチウィンドウプログラム1に基づきマルチウィンドウシステムを構築するCPU3を備える。マルチウィンドウプログラム1は、画面情報を保持する画面情報保持部1aと、画面遷移制御部1bと、画面表示部1cと、画面遷移テーブル1dとを含む。画面遷移テーブル1dには、重なるように遷移するダイアログ群の階層構造を記述した階層構造テーブル(図9に示すTA1)と、ユーザから入力デバイス4を介して入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述した遷移先テーブル(図9に示す)TA2とが含まれている。画面遷移制御部1bは、画面遷移テーブル1dに基づきダイアログ間の画面遷移を自動的に行うように遷移先のダイアログを検索し、画面表示部1cは検索されたダイアログを表示手段6に表示するように制御する。

【0034】次に、本実施の形態における画面遷移制御について図2を参照しながら説明する。図2は図1の情報処理装置上に構築されているマルチウィンドウにお

る画面遷移制御を実現するための構成を示す概念図である。

【0035】本実施の形態では、画面遷移を自動的に行うために必要な関数として3つの外部公開関数を用意し、画面遷移テーブルを作成した後に上記関数をアプリケーション内で呼び出すことにより画面遷移の自動化を実現している。

【0036】具体的には、画面遷移エンジンには、画面遷移を開始する関数、画面を遷移させる関数、画面遷移テーブルに記述された階層構造を無視して特定のダイアログを一番前面に表示させる関数の3つの関数が上記外部公開関数として保持されている。ダイアログ(画面)上でユーザからイベントが入力されると、該入力されたイベントに対応する外部公開関数が呼び出され、画面遷移エンジンでは、画面遷移テーブルを参照して呼び出された関数に応じた画面遷移を行うようにダイアログの作成、破棄を行う。

【0037】次に、画面遷移の手順について図3を参照しながら説明する。図3は図1の情報処理装置上に構築されているマルチウィンドウシステムにおける画面遷移の手順を示すフローチャートである。

【0038】標準画面上でボタンが押下されると、図3に示すように、まずステップS31において、画面遷移テーブルを参照して、押下されたボタンによって次に表示すべき画面を検索する。次いで、ステップS32に進み、押下されたボタンが終了ボタンであるか否かを判定し、押下されたボタンが終了ボタンであれば、標準画面に戻る。押下されたボタンが終了ボタンでなければ、ステップS33に進み、画面を破棄し、重ねて表示し、または画面を入れ替わるように表示する。次いで、再度ステップS31に戻り、表示されている画面上で押下されたボタンに対応する画面の検索を行う。そして、表示されている画面上で終了ボタンが押下されると、標準画面に戻り、終了ボタン以外の他のボタンが押下されると、この押下されたボタンに対応する画面遷移が行われる。

【0039】次に、画面遷移開始の手順について図4を参照しながら具体的に説明する。図4は図1の情報処理装置のマルチウィンドウシステムにおける画面遷移開始の手順を示す図である。

【0040】画面遷移を開始する際には、まず、ダイアログ上でボタンが押下されてイベントが発生し、上述した画面遷移を開始する関数が呼び出される。画面遷移エンジンでは、この関数の呼出により画面遷移テーブルを参照して画面遷移を開始し、そのグループ内でデフォルトで起動するダイアログを作成して画面上に表示する。ここで、既に他の画面遷移が行われている途中で画面遷移を新規に行う場合には、既に展開されたダイアログを全て破棄して終了し、その後新たな画面遷移を開始する。

【0041】この画面遷移が一旦開始されると、ユーザ

から入力されたイベントに対して画面遷移テーブルの内容に従って遷移が繰り返される。ユーザから入力されたイベントに対して画面を遷移させる場合には、そのイベントハンドラの中で画面を遷移させる関数の呼出を行う。また、どのコントロール（イベントを発生したGUI部品）からコールされたかを画面遷移エンジンに知らせる必要があるが、これはコントロールID (Ctrl ID) を引数で渡すことにより識別される。すなわち、このCtrl IDの値を用いてテーブル上から次に遷移する画面を検索する。この関数を使用する場合の注意点としては、この関数を呼び出したときにテーブル内に記述されている遷移先の指定によっては、呼び出したダイアログ自身を破棄することもありえる。

【0042】このダイアログの破棄を含む画面遷移の手順を図5を参照しながら具体的に説明する。図5は図1の情報処理装置のマルチウィンドウシステムにおける画面遷移の手順を示す図である。

【0043】ユーザの入力に対するイベントが発生すると、図5に示すように、イベントを発生したGUI部品から画面を遷移させる関数の呼出が行われる。この呼出には、コントロールID (Ctrl ID) が用いられ、その引数が画面遷移エンジンに渡される。画面遷移エンジンでは、このCtrl IDの値を用いてテーブル上から次に遷移する画面を検索する。この関数によりダイアログ自身を破棄する場合には、このダイアログに対して破棄の確認をFLOW_DESTROYメッセージを用いて行う。この場合には、親ダイアログに子ダイアログの破棄をFLOW_CHILDDESTROYメッセージを用いて通知する。ここで、破棄を中止する場合には、CALLBACKDATA構造体を用いられる。このCALLBACKDATA構造体については、後述する。

【0044】これに対し、遷移先のダイアログが見出されると、遷移先への画面遷移が行われる。画面遷移では、ダイアログが作成され、作成されたダイアログが重なるようにまたは入れ替わるように表示される。

【0045】次に、モーダルダイアログを用いてダイアログのネスティングを行う場合について図6および図7を参照しながら説明する。図6は通常のネスティングフローによる画面遷移を示す図、図7は図1の情報処理装置のマルチウィンドウシステムにおけるネスティングフローによる画面遷移を示す図である。

【0046】モーダルダイアログを用いてダイアログのネスティングフローを実行する場合には、図6に示すように、通常、モーダルダイアログを作成し、このダイアログプロシージャの中でもう一度モーダルダイアログを作成するという手順を取る。

【0047】この通常の手順においては、例えば、ダイアログ1からダイアログ3への遷移が生じた場合に、ダイアログ2の中でダイアログを作成する必要がある。従って、この場合にはダイアログ2のダイアログプロシージャの中でダイアログ3を作成するというコードを記述

しなければならない。これに対し、本実施の形態では、図7に示すように、ネスティングするダイアログを全てモードレスダイアログで作成し、ダイアログ1から全てのダイアログを制御可能なようにしている。よって、上述したようなハードコーディングを混入させることなく、ダイアログのネスティングフローを実現することができる。また、図7に示す例では、ダイアログ3の表示期間中、ダイアログ2を無効にしなければならないから、このダイアログを無効にする機能を画面遷移エンジンの中で実現している。

【0048】ここで、ユーザのイベントに対する画面遷移の具体的な処理の手順を図8および図9を参照しながら説明する。図8は図1の情報処理装置のマルチウィンドウシステムにおけるダイアログの生成、破棄に関する処理の手順を示すフローチャート、図9は画面を遷移させる関数内部における処理手順を示すフローチャート、図10はCALLBACKDATA構造体の構成を示す図である。なお、本説明では、画面遷移エンジンが起動されてダイアログが表示された状態を想定している。

【0049】図8を参照するに、まずステップS141においてイベントの入力を待ち、この入力されたイベントがダイアログ上のボタン押下によるメッセージであるときには、ステップS143に進み、メモリ5に格納されているダイアログ情報の読出し、書込みを行い、続くステップS144で、画面を遷移させるための画面遷移関数の呼出を行う。この画面遷移関数の呼出により該関数内部の処理に移行する（図9に示す）。

【0050】この呼び出した画面遷移関数内部では、図9に示すように、まずステップS151において、画面遷移テーブル1dに含まれる遷移先テーブルT A2と階層構造テーブルT A1とを参照して、次に表示すべきダイアログを検索する。続くステップS152では、階層構造テーブルT A1を参照して検索により得られた次に表示させるダイアログ群と現在表示されているダイアログ群とを比較し、破棄すべきダイアログと生成すべきダイアログとをリストアップする。そしてステップS153に進み、破棄すべきダイアログが存在するか否かを判定する。破棄すべきダイアログが存在しないときには、ステップS154に進み、生成すべきダイアログが存在するか否かを判定し、生成すべきダイアログが存在する場合には、ステップS155に進み、生成すべきダイアログを生成し、そして再度上記ステップS154に戻る。これに対し、生成すべきダイアログが存在しないときには、本処理を終了する。

【0051】ステップS153において破棄すべきダイアログが存在すると判定されたときには、ステップS158に進み、まず破棄するダイアログに対して破棄の確認を行う。そして処理が図8に示すダイアログ破棄通知に関するイベントハンドラへ移行する（上記ステップS141）。図8に示すステップS141において破棄通

知が確認されると、ステップS145に進み、メモリ5に格納されているダイアログ情報の読み出し、書き込みを行う。この処理は必要に応じて行われる処理である。次いで、ステップS146に進み、ダイアログが持つ内部情報に基づきダイアログを実際に破棄するか否かを判定する。ここで、ダイアログを実際に破棄するときには、破棄を行い、上記ステップS141に戻る。これに対し、ダイアログの破棄を何らかの理由で中止すると判定すると、ステップS147に進み、破棄の通知に際して、画面遷移エンジンから送信されたCALLBACKDATA構造体(図10に示す)の中の破棄中止フラグIntrptFlgに破棄中止を促す値をセットする。このCALLBACKDATA構造体は、アプリケーションの都合で、現在のダイアログの遷移のために破棄することを中止する場合に、画面遷移エンジンに対して破棄中止を通知するための構造体である。例えば、図18に示すように、IDD_DLG5からIDD_DLG3への遷移を行う際、通常ならば、IDD_DLG5、IDD_DLG4、IDD_DLG2のダイアログが順番に破棄される。しかし、各ダイアログが破棄される直前にCALLBACKDATA構造体の中の上記破棄中止フラグIntrptFlgをセットすることによりそれ以降の破棄を中止することができる。また、例えば、IDD_DLG4の段階で破棄を中止すれば、IDD_DLG4、IDD_DLG2の各ダイアログの破棄は行われず、以降IDD_DLG4のダイアログ画面のイベント待ちとなる。

【0052】この機構を用いることにより、画面遷移を行う際に任意の処理(すなわちステップS145の処理)を行うことができ、またアプリケーションが保持している条件によって、ダイアログの破棄を確認し(ステップS146)、破棄を中止することもできる(ステップS147)。また、画面遷移関数内部では、図9に示すように、ステップS159においてCALLBACKDATA構造体の中の上記破棄中止フラグIntrptFlgがセットされているか否かを判定し、上記破棄中止フラグIntrptFlgがセットされていなければ、ステップS160に進み、ダイアログを破棄し、上記ステップS153に戻る。このダイアログの破棄に伴い図8に示すステップS141に移行し、該ステップにおいてダイアログ破棄通知があっ*

*たことを判定する。そしてステップS142に進み、メモリ5に格納されているダイアログ情報の読み出し、書き込みを行う。この処理は必要に応じて行われる処理である。

【0053】次に、画面遷移テーブル1dを作成する方法について図11ないし図15を参照しながら説明する。図11ないし図15は図1の情報処理装置のマルチウィンドウシステムに用いられている画面遷移テーブルの作成方法を説明するための図である。なお、ここでは、図42に示すダイアログの階層構造をテーブル化する場合を例に説明する。

【0054】本実施の形態では、画面遷移を自動化するために、2つのテーブルが重要となる。すなわち、図11に示すような各ダイアログに対して、各ダイアログの階層構造を表現することができる階層構造テーブルTA1と、各ダイアログ毎に遷移先を定義することができる遷移先テーブルTA2との2つのテーブルが用いられる。各ダイアログに対するテーブル化を実際に行うためには、図12に示すように、各ダイアログに対して具体的な条件設定を付加する。具体的には、各ダイアログに対しては、そのダイアログにIDD_DLGで始まるリソースID(識別子)が割り当てられる。例えば、ダイアログ1に対しては、IDD_DLG1が割り当てられる。また、それぞれのダイアログには、動作を定義するダイアログプロシージャDlgProc(プログラム)がバインドされる。各ダイアログに対しては、固有の識別番号が割り当てられており、この識別番号は1つの階層構造内でダイアログユニークであればその順番は問われない。ただし、「0」の番号を割り当てることは許されない。これは、後述するように、「0」の番号が遷移終了を指定するための識別番号として使用されるためである。

【0055】ダイアログベースのアプリケーションにおけるダイアログ間のつながりは、基本的に所有関係である。このような構成における階層構造は、次の表1に示すようなテーブルで表される。

【0056】

【表1】

ダイアログの識別番号	ダイアログリソースID	親ダイアログの番号	ダイアログごとの遷移先情報	グループ識別番号
1	IDD_DLG1	0	遷移先テーブル1	1
2	IDD_DLG2	1	遷移先テーブル2	1
3	IDD_DLG3	1	遷移先テーブル3	1
4	IDD_DLG4	2	遷移先テーブル4	1
5	IDD_DLG5	4	遷移先テーブル5	1

ここで、上述の表1には、階層構造を表現するための必要な項目として、ダイアログの識別番号、ダイアログID、親のダイアログ(オーナーウィンドウ)の識別番号、ダイアログ毎の遷移先情報、グループ番号(後述参照)

が含まれ、この遷移先情報は、入力されたイベントによってどのダイアログを表示させるかを定義した遷移先テーブルの先頭アドレス(後述参照)である。この表1で表されたテーブルは、実装段階で各プログラミング言語

の構成体を用いて表現される。

【0057】この階層構造テーブルを利用して画面遷移を開始する場合、先頭の配列要素には、一番最初に起動させるダイアログのデータを記述する。

【0058】次に、表1内で「遷移先テーブル1」と記述している、入力されたイベントに対する遷移先のテーブル定義について説明する。

【0059】ここでは、図12に示す識別番号1のダイアログには、図13に示すように、3つのボタンIDC_SET1、IDC_SET2、IDC_CANCELが配置されており、各ボタンにはそれぞれ固有のIDが割り当てられているとする。このボタンのIDはウィンドウリソースの識別子であ *

	コントロールID	遷移先ダイアログの識別番号
IDC_SET ボタン	IDC_SET1	2
IDC_SET2 ボタン	IDC_SET2	3
IDC_CANCEL ボタン	IDC_CANCEL	0

なお、遷移先のダイアログ識別番号(ID)を「0」とした場合、現在行っている画面遷移制御を中断し、現在表示されているダイアログを全て破棄し、画面遷移を終了するように設定している。また、終了時に複数のダイアログが重なって表示されている場合には、前面に配置されているダイアログから順に破棄する。上記表1内の項目「ダイアログごとの遷移先情報」は、この表2のテーブルの先頭アドレス(C言語でいえば、遷移先テーブルの構造体のポインタ)を格納し、これにより、それぞ

ダイアログリソースID	ダイアログプロシージャの先頭アドレス
IDD_DLG1	ダイアログ1のプログラム(DlgProc1とする)の先頭アドレス
IDD_DLG2	ダイアログ2のプログラム(DlgProc2とする)の先頭アドレス
IDD_DLG3	ダイアログ3のプログラム(DlgProc3とする)の先頭アドレス
IDD_DLG4	ダイアログ4のプログラム(DlgProc4とする)の先頭アドレス
IDD_DLG5	ダイアログ5のプログラム(DlgProc5とする)の先頭アドレス

次に、ハイパーリンク機能とシーケンシャルフローとについて図16ないし図19を参照しながら説明する。図16ないし図18は図1の情報処理装置のマルチウィンドウシステムにおけるハイパーリンク機能を説明するための図、図19は図1の情報処理装置のマルチウィンドウシステムにおけるシーケンシャルフローを説明するための図である。

【0063】上記図11に示す階層構造を有するダイアログにおいて、現在識別番号5のダイアログが操作可能であるとすると、画面は図16に示すように重なり合った状態にある。ここで、従来の方法では、図16に示す状態において、ダイアログがユーザによって1枚ずつ破棄される。これに対し、本実施の形態では、識別番号5のダイアログの遷移先テーブル内に、オーナウィンドウ以外のウィンドウを定義することなどによって、識別番号1、または識別番号3のダイアログなどに直接ジャンプすることができ、また途中でそれを取り消すことがで

*り、この値はユーザがどのボタンを選択したかの識別に利用される。すなわち、図12に示す識別番号1のダイアログに対して、図13に示すように、IDC_SET1が選択された場合識別番号2のダイアログ(図14に示す)へ、IDC_SET2が選択された場合識別番号3のダイアログ(図15に示す)へそれぞれ遷移し、そしてIDC_CANCELが選択された場合にはこの遷移を終了する、という条件が設定されていることになる。この遷移先を定義するためのテーブルの記述例を表2に示す。

【0060】

【表2】

※れのダイアログ上のボタンと遷移先とが関連付けられる。

【0061】また、ダイアログIDと、ダイアログの動作を定義するダイアログプロシージャ(プログラム)とが表3に示すテーブルにより関連付けられ、このテーブルには、ダイアログID、ダイアログプロシージャの先頭アドレスの各項目が含まれる。

【0062】

【表3】

きる。このように、他の画面に一気に遷移させ、また途中で取り消す機能をハイパーリンク機能といい、このハイパーリンク機能による画面遷移例を図17および図18に示す。このような遷移先を指定するためには、遷移先テーブルTA2に次に表示したいダイアログの識別番号を記述すればよい。

【0064】また、ハイパーリンク機能を利用してシーケンシャルフローを実行することが可能である。例えば、図19に示すように、図11に示す階層構造のダイアログにおいてダイアログ2が操作可能な状態からイベントにより識別番号3のダイアログに遷移したときには、最前面のダイアログが入れ替わったように見える。すなわち、シーケンシャルフローが実行されたことになる。

【0065】このように、本実施の形態では、重なるように遷移するダイアログ群の階層構造を記述した階層構造テーブル(図9に示すTA1)と、ユーザから入力デ

パイプ4を介して入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述した遷移先テーブル(図9に示す)TA2とを含む画面遷移テーブル1dに基づき、ダイアログ間の画面遷移を自動的に行うように遷移先のダイアログを検索し、検索されたダイアログを表示手段6に表示するように制御するから、ネスティングフロー、シーケンシャルフローを含む画面遷移を効率良く実行することができる。

【0066】このようなマルチウィンドウシステムにおいて、画面遷移テーブル1d内で同一のダイアログが何度も遷移に関わる場合には、そのダイアログが必要とする資源を共有し、その遷移先だけを複数定義する。すなわちダイアログデータを共有化する。

【0067】このダイアログデータの共有化について図20および図21を参照しながら説明する。図20は図1の情報処理装置のマルチウィンドウシステムにおけるダイアログデータの共有例を示す図、図21はダイアログの共有における画面遷移の考え方を示す図である。

【0068】ここでは、図20に示すように、識別番号3のダイアログが識別番号2のダイアログおよび識別番号1のダイアログをオーナウィンドウとしてしている場合を考える。すなわち、識別番号3のダイアログが操作可能な状態になる画面遷移としては、識別番号1のダイアログに識別番号3のダイアログが重なる場合と、識別番号2のダイアログに識別番号3のダイアログが重なる場合との2種類のケースがある。そして、その遷移は識別番号2、3のダイアログの遷移先テーブルによって定*

ダイアログの識別番号	ダイアログリソースID	親ダイアログの番号	ダイアログごとの遷移先情報	グループ識別番号
1	IDD_DLG1	0	画面遷移テーブル1	1
2	IDD_DLG2	1	画面遷移テーブル2	1
3	IDD_DLG3	2	画面遷移テーブル3	1
4	IDD_DLG3	1	画面遷移テーブル4	1

すなわち、本実施の形態では、階層構造テーブル上で識別番号3のダイアログを仮想的に別々のものと解釈し、その遷移先テーブルを遷移先テーブル3と遷移先テーブル4とでそれぞれ指定可能にしている。また、ウィンドウリソースを表す識別子IDD_DLG3は、同じものを使用しているから、デザインデータおよび画面遷移の情報を分離されたダイアログ3に対してアプリケーションコードはそれぞれ全く同一のものを利用することができる。

【0072】このように、ダイアログデータを共有化することによって、画面遷移を実現するためのリソースを必要最小限にすることができ、ひいてはメモリ容量の増加などのメモリへの負担を軽減させることができる。

【0073】また、本実施の形態においては、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述することが可能である。このグループ化に

*義されており、識別番号1から識別番号2のダイアログへ、また識別番号1から識別番号3のダイアログへネスティングフローする場合には何ら問題は生じない。しかし、識別番号3のダイアログに対して外部からイベントが入力されたときに、どのような画面遷移を経たかによって次に表示する画面を変更したい場合がある。

【0069】具体的に説明すると、今、識別番号3のダイアログが操作可能な状態で、ダイアログ上には、キャンセルボタンがあるとする。このキャンセルボタンの機能としては現在の設定をキャンセルしてダイアログを破棄するものである。しかし、上述の画面遷移テーブルの記述方法では、1つのイベントに対して1つの遷移先しか設定することができないから、識別番号2のダイアログから識別番号3のダイアログへ遷移した場合には、識別番号2のダイアログへ、識別番号1のダイアログから識別番号3のダイアログへ遷移した場合には、識別番号1のダイアログへというような遷移を1つの遷移先テーブルの中に記述することはできない。

【0070】そこで、ダイアログの階層構造を図21に示すように考えることにより、上述の事態を回避している。具体的には、今、識別番号3のダイアログを識別番号1のダイアログに対するものと識別番号2に対するものとそれぞれ別個に考え、それぞれを階層構造テーブルにセットする。これにより階層構造テーブルは表4に示すように記述することができる。

【0071】

【表4】

ついて図22ないし図26を参照しながら説明する。図22は図1の情報処理装置のマルチウィンドウシステムにおける複数のダイアログを共有する例を示す階層構造図、図23は図22のダイアログをグループ化した例の階層構造を示す図、図24は図22のダイアログ12以下をグループ化した例の階層構造を示す図、図25は複数のダイアログをグループ化した場合のグループ内のボタン押下に対する動作を示す図、図26はグループの破棄例を示す図である。

【0074】ダイアログベースのアプリケーションの画面遷移では、1つの機能を複数の画面で設定することが多い。例えば、図22に示すように、ダイアログ12以下の階層構造が共有化されている。しかし、本例では、ダイアログの共有化で述べたと同様に、共有されているダイアログ12がダイアログ5の前面に重なった場合と

ダイアログ11の前面に重なった場合とでは、その後の画面遷移が異なる。そこで、ダイアログデータの共有化をする方法と同様に、ダイアログ12以下を仮想的に別々のものであると考える。これにより同一のウィンドウリソースを用いて階層構造上の共有を表現することが可能になる。さらに、このダイアログ群を1つにまとめてグループ化する機構を導入する。ダイアログ群をまとめてグループ化するメリットは、グループ内に所属するダイアログの遷移先を任意に変更することができるという*

*ことである。すなわち、例えば図22に示すダイアログ11から遷移の違いによってその遷移先を個別に設定することが可能になる。

【0075】これらのことを考慮すると、図22に示す階層構造は、図23、図24にそれぞれ示す階層構造に書き直すことができ、各階層構造をそれぞれテーブル化したものを表5、表6に示す。

【0076】

【表5】

ダイアログの識別番号	ダイアログリソースID	親ダイアログの番号	ダイアログごとの遷移先情報	グループ識別番号
1	IDD_DLG1	0	遷移先テーブル1	1
2	IDD_DLG2	1	遷移先テーブル2	1
3	IDD_DLG3	1	遷移先テーブル3	1
4	IDD_DLG4	2	遷移先テーブル4	1
5	IDD_DLG5	2	遷移先テーブル5	1
6	IDD_DLG6	2	遷移先テーブル6	1
7	IDD_DLG7	3	遷移先テーブル7	1
8	IDD_DLG8	3	遷移先テーブル8	1
9	IDD_DLG9	4	遷移先テーブル9	1
10	IDD_DLG10	7	遷移先テーブル10	1
11	IDD_DLG11	7	遷移先テーブル11	1
12	0(グループなので特定のIDは持っていない)	5	遷移先テーブル12	2
13	0(グループなので特定のIDは持っていない)	11	遷移先テーブル13	2

【0077】

※ ※【表6】

ダイアログの識別番号	ダイアログリソースID	親ダイアログの番号	ダイアログごとの遷移先情報	グループ識別番号
1	IDD_DLG_C1	0	遷移先テーブル2-1	2
2	IDD_DLG_C2	1	遷移先テーブル2-2	2
3	IDD_DLG_C3	1	遷移先テーブル2-3	2

表5に示すテーブルにおいて、グループ1では、ダイアログ識別番号12、13にグループ2のダイアログ群を割り当てられている。また、グループ1内のいずれかのダイアログからグループ2へ、すなわち識別番号12または識別番号13のダイアログへ遷移した場合には、最初にグループ2の階層構造を表すテーブルの内の一番先頭に記述されているダイアログが起動するように設定されている。ここで、グループ識別番号の項目には、普通のダイアログの場合は、そのダイアログが所属するグループの番号を、グループ化されたもの場合には、そのグループの番号を指定する。

【0078】グループ2の階層構造の中で記述されているダイアログの識別番号は、グループ内でユニークであ

ればよく、グループ1の中で使用されている番号と重複してもよいが、遷移先テーブルはアプリケーションでユニークする必要があるから、グループ1内で使われている構造体名と重複させることは避けなければならない。

40 【0079】なお、グループの識別番号の登録のために、テーブルを1つ作成し、ここで使用しているグループの識別番号は表7のグループテーブル内で定義された番号を使用する。このグループテーブルの項目は、グループの識別番号、階層構造テーブルの先頭アドレスの順である。

【0080】

【表7】

	グループの識別番号	階層構造テーブルの先頭アドレス
グループ1	1	GROUP1
グループ2	2	GROUP2

次に、グループ化を考えた場合における遷移先テーブルの書式について説明すると、図22に示す例では、グループ2をグループ1から呼び出している。そして、グループ1からグループ2へ遷移が行われた場合には、グループ2の各ダイアログに対して定義されている遷移先テ

ーブルの記述に従って画面遷移が行われる。しかし、グループ1内のどのダイアログからの遷移かによって、グループ2内のダイアログの遷移をカスタマイズする場合が多い。
【0081】このような場合に対しては、次のようにカスタマイズする。図25に示すダイアログの階層構造においてはグループ2内のダイアログ2-2上にボタンが1つあるとし、このダイアログ2-2の遷移先テーブルではボタンが押下された場合にグループ2内のダイアログ2-1に遷移するように記述されているものとする。ここで、グループ1側から見て識別番号1.2としてグループ2内に遷移した場合と識別番号1.3としてグループ2に遷移した場合とで、グループ2内のダイアログ2-*

	コントロールID	遷移先ダイアログの識別番号
	1	0
IDC_BTN1 ボタン	IDC_BTN1	1

従って、グループ1内の遷移と関連させてグループ2のダイアログ2-2のボタンIDC_BTN1をカスタマイズしないときには、この遷移先テーブルに従って遷移動作が行われる。そして、グループ1内のダイアログ5をオーナーとしているグループ2（識別番号1.2が割り当てられている）においては、グループ2内のダイアログ2-2上※

	コントロールID	遷移先ダイアログの識別番号
	1	0
IDC_BTN1 ボタン	IDC_BTN1	5

ここで、この記述がないときには、グループ2内のダイアログ2-2のデフォルトの遷移先テーブル（遷移先テーブル2-2）内で定義されている遷移先が参照される。このように遷移先をカスタマイズしてあれば、カスタマイズした遷移先の方が優先的に参照される。また、遷移先のカスタマイズは遷移するグループの中のどのコントロールIDに対しても可能である。このため、ダイアログ作成者は、遷移先がカスタマイズされる可能性があるコントロールに割り当てたIDが同一グループ内で★

	コントロールID	遷移先ダイアログの識別番号
	0	0

* 2上のボタンが押下されたときの遷移先をカスタマイズすることを考える。このグループ2内のダイアログ2-2上のボタンのIDをIDC_BTN1とし、このボタンIDC_BTN1が選択された場合には、デフォルトでグループ2のダイアログ2-1へ遷移するように設定されている。そして、グループ1内で識別番号1.2のグループ2として遷移している場合には、ボタンIDC_BTN1の押下に対する遷移先をダイアログ5に、識別番号1.3のグループ2として遷移している場合には、ボタンIDC_BTN1の押下に対する遷移先をダイアログ11になるようにカスタマイズした例を示す。

【0082】まず、ダイアログの行き先を定義する、図25に示すダイアログの階層構造におけるグループ2内のダイアログ2-2に対する遷移先テーブルを表8に表す。このテーブルがデフォルトの遷移先テーブルとなる。

【0083】

【表8】

※でボタンIDC_BTN1が選択された場合、グループ1内のダイアログ5へ遷移するように変更するため、グループ1の識別番号1.2の遷移先テーブルにカスタマイズする行き先を表9に表すように記述する。

【0084】

【表9】

★重複しないように注意する必要がある。なお、カスタマイズする必要がなくても遷移先テーブルは必ず作成しなくてはならず、表10に示すように、「0」個の要素があると記述する。同様に、グループ1内のダイアログ番号1.3に割り当てられているグループ2に対する遷移先を定義する遷移先テーブルは表11に示すように記述される。

【0085】

【表10】

21

	コントロールID	遷移先ダイアログの識別番号
	1	0
IDC_BTN1 ボタン	IDC_BTN1	11

22

さらに、グループ化、共有化に際しては、いくつかの注意点があるので、それについて説明する。

【0087】上記図25に示すダイアログの階層構造においてグループ2のダイアログ2-2のボタンIDC_BTN1が押下されたときに遷移を終了するように指定する場合を考える。すなわち、遷移先テーブルで行き先のダイアログの識別番号に「0」を指定する場合を考える。通常、遷移が複数のグループ間に亘っていない場合には、現在表示されているダイアログの全てを破棄して終了する。これに対して、図26に示すように、グループ1とグループ2との間に階層関係があり、グループ2の中の遷移で終了命令が現れた場合には、そのグループに属す*

*るダイアログだけが破棄される。仮にここで呼んだ側のグループ1に属するダイアログの全てを破棄して終了するようにしたいときには、グループ1の中でグループ2をダイアログ識別番号12として定義しているが、この遷移先を定義する遷移先テーブルの中でグループ2内のダイアログ上のボタンのIDの遷移先をカスタマイズしてグループ1内での遷移の終了（すなわち行き先は「0」）を指定するようにすればよい。これを実際の記述すると、表12のようになる。

【0088】

【表12】

	コントロールID	遷移先ダイアログの識別番号
	1	0
IDC_BTN1 ボタン	IDC_BTN1	0

次に、ダイアログを破棄する際のメッセージ通知機構について図27および図28を参照しながら説明する。図27はダイアログの破棄の一例を示す図、図28はメッセージ通知機構の処理手順を示す図である。

【0089】ここでは、図27に示すように、識別番号2のダイアログ上のボタンの押下により、識別番号2のダイアログが破棄されて識別番号1のダイアログのみが残った場合の画面遷移を例に説明する。

【0090】ダイアログが破棄される際には、図28に示すように、まず、画面遷移関数から対応するダイアログを実際に破棄してよいかの確認の要求に応じて、ダイアログに対して画面遷移エンジンからFLOW_DESTORYメッセージが送信され、これに対応したハンドラの中でダイアログの終了処理が行われる。また、この際に上述したCALLBACKDATA構造体（図10に示す）を用いて破棄時の情報の受渡しが行われると同時に、IntrptFlagの値をダイアログプロシージャから設定することによって、そのダイアログを破棄するかどうかを設定される。仮に何かの条件によりダイアログの破棄を中断したい場合には、IntrptFlagがTRUEに設定される。また、CALLBACKDATA構造体中のReturnIDは画面遷移関数の引数として指定することができる、例えばOK、CANCEL、GO BACKなど予め定義した値のいずれかを使用することによりそれぞれのダイアログが破棄されるときにその破棄を中断するか否かの判定に使用することができる。そして、その破棄が終了すると、そのダイアログのオーナーウィンドウにFLOW_CHILDDESTORYメッセージが送信され、自身が所有しているウィンドウが破棄されたことを知ることができる。また、必要であればそのメッセージハンドラの中で何らかの処理が行われることになる。この確認は、一度の遷移

で全てのダイアログに送信され、その途中で破棄の中断が確認されると、その時点で破棄を中断しその画面からの遷移が続行される。

【0091】本実施の形態では、画面遷移テーブル1dを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力されたユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄することができる。この機構について図29を参照しながら説明する。図29は図1の情報処理装置のマルチウィンドウシステムにおける動的に設定したダイアログの表示例を示す図である。

【0092】応用的なダイアログベースのアプリケーションの機能のひとつとして、深い階層に位置するダイアログを登録しておき、そのダイアログだけを呼び出すという機能がある。しかし、一度画面遷移を開始すると、基本的に画面遷移テーブル1dを参照して遷移を実現させているために、ユーザが画面遷移の途中で登録した動的な画面遷移テーブルに基づきダイアログを表示させることは困難である。そこで、その機能を実現するために、ウィンドウの階層構造を一時的に無視して特定の画面を表示可能なように構成する。

【0093】この機能により、ダイアログプロシージャの再利用性を向上するというメリットが得られる。通常のダイアログプロシージャは画面遷移を行わせるから、全てのボタン操作に対するイベントハンドラの中で画面遷移関数をコール可能なコードになっている。ところがこれでは、いずれかのダイアログを表示させてどこにも遷移させずにダイアログを閉じるような操作を行いたい場合、通常使用しているダイアログプロシージャをそのまま使用することができず、通常使用しているものとは

別のものを作成しなければならない。そこで、通常使用しているダイアログプロシーダをそのまま利用することができるよう、新たに特定のダイアログを画面遷移テーブル1dに記述された階層構造を無視して最前面に表示させる関数を準備する。この関数は、その関数をコールしたスレッド上で遷移している画面の最前面に指定したダイアログを表示させるためのものである。すなわち、上記関数は、表示されているダイアログのダイアログプロシーダの中で画面遷移関数がコールされたとしても、画面遷移を行わずにその画面を破棄させる。従って、通常使用しているダイアログプロシーダがそのまま利用することができ、開発効率の向上、メモリの節約につながる。例えば、図29に示すように、この関数は、一時的なダイアログDLG_TMPを表示させ、その後、そのダイアログを破棄した場合に、以前に遷移していた画面の状態をそのまま続行することができる。

【0094】画面遷移を行う上で必要になる情報は、マルチスレッドをサポートしたOS上でそれぞれのスレッド上で独立して画面遷移を実現することができるように、スレッドIDをキーにして管理する。この管理方法について図30ないし図36を参照しながら説明する。図30は図1の情報処理装置のマルチウィンドウにおけるダイアログの階層関係と内部情報の保持方法とを示すための図、図31は図30の内部情報を保持する構造体を示す図、図32は図30の内部情報の推移を説明するための画面遷移例を示す図、図33ないし図36は内部情報の推移を表す図である。

【0095】上述の管理方法では、図30に示すように、ダイアログの階層構造をスタック状の情報構造に置き換えて保持する。具体的には、図31に示すように、FLOWMEMMGR構造体がそれぞれのフローに対応して作成され、そしてフローが終了すると、破棄される。また、この構造体は、それぞれにFLOWSTACK構造体を有し、FLOWSTACK構造体に積まれた情報をダイアログの階層構造と対応させることにより、ダイアログのネスト状況や、それぞれのダイアログ固有の情報をスタック状に管理するようにしている。

【0096】ここで、図31に示すFLOWMEMMGR構造体、FLOWSTACK構造体のメンバの意味は次の通りである。

FLOWMEMMGR:

DWORD ThreadID; スレッドID

TREETABLE *TreeTable; Tree情報が格納された構造体へのポインタ

FLOWSTACK *FlowStack; FLOWSTACK構造体配列(スタック)のポインタ

HWND RootWnd; フローを開始したウィンドウのハンドル

WORD StackNum; 現在スタックにいくつのデータが積まれているかを表すカウンタ

BOOL fCallDlg; CallDialogが使用されている場合のフラグ

FLOWSTACK:

WORD DlgNum; ダイアログ識別番号

WORD DlgID; ダイアログリソースID

HWND hWnd; ダイアログのウィンドウハンドル

HWND hWndParent; 親のウィンドウハンドル

BOOL Check; ウィンドウが破棄されるかどうかをチェックするフラグ

WORD TreeNum; 現在遷移中のテーブルのグループ番号

WORD GroupNum; 分岐元のテーブル内でのグループ番号

10 TREETABLE *CurrentTable; ダイアログ識別番号に対応したTREETABLE構造体のポインタ

TRRETTABLE *PrevTable; 分岐元グループのテーブル内でのグループ番号に対応したTREETABLE構造体のポインタ

次に、実際の画面遷移の発生に伴い実行される画面遷移関数の内部における処理について説明する。ここでは、図32に示すように、ダイアログ4からダイアログ3へ画面が遷移する場合を例に示す。なお、以下に示す具体例は既に説明した図8および図9のフローチャートに対応する。

20 【0097】まず、画面遷移を開始する関数をフロー開始のダイアログから起動する。上述したように、1つのスレッドに対して1つのフローしか展開することができないから、1つのフローを展開中に別のフローを開始したときには、既に起動していたフローを終了して新たなフローを開始するものとする。なお、フローが開始されると、スレッドIDをキーにセットされるとともに、FLOWMEMMGR構造体のひとつに値がセットされる。そして、1つのダイアログが起動し、その構造体に対してFLOWSTACK構造体も作成され、最初のダイアログ情報が格納される。

【0098】最初のダイアログが作成されると、次からは画面を遷移させる関数を呼び出すことになる。この関数の内部では、下記のような処理が行われる。

【0099】すなわち、ユーザからのイベントが入力されると(ステップS141)、今操作が行われているダイアログのスレッドIDを取得し、そのスレッドIDをキーにボタンを操作したダイアログに関するスタック情報を取得する。この取得した情報に基づき、階層構造テーブルおよび遷移先テーブルから次に画面に表示するダイアログを検索する(ステップS151)。ただし、遷移先テーブルに指定されている、コントロールに対する行き先のダイアログ番号に「0」が指定されている場合、全てのダイアログを破棄して画面遷移を開始したダイアログに戻る(ステップS141)。

【0100】どのダイアログを画面に表示させるかを決定すると、図33に示すように、そのダイアログまでの階層関係を辿りながらその階層関係を表すための別のスタックを構築する(ステップS152)。

50 【0101】次いで、別のスタックに作成されたものと

現在のスタックとを比較して、どちらのスタックも存在する共通なダイアログがある場合、図34に示すように、現在のスタックのそのダイアログに対応する構造体のCheckメンバにTRUEをセットする（ステップS152）。

【0102】そして、現在のスタックにおいてCheckが付いていないダイアログは破棄しなければならないので、これらを破棄する（ステップS153、ステップS160）。ただし、図35に示すように、上述したように、破棄する際にそのダイアログに確認を取るから、破棄を中断するフラグ（ステップS158、ステップS159）が立っているときには、そこで画面遷移は終了し、次のボタンが押下されるなどのイベントの入力待つ（ステップS141）。

【0103】破棄すべきダイアログがなかった場合（ステップS153）には、図36に示すように、この時点で共通なダイアログ以外は破棄されているはずであるから、次に新たに作成されたスタックから共通のダイアログ以降のダイアログを1つずつ作成して重ねていき（ステップS155）、同時に現在のスタックにその情報をプッシュしていく。

【0104】そして、再度、この作成されたダイアログから画面を遷移させる関数を呼び出す処理へ移行する（ステップS141）。

【0105】

【発明の効果】以上に説明したように、請求項1記載のマルチウィンドウシステムによれば、重なるように遷移するダイアログ群の階層構造を記述したテーブルと、ユーザから入力されたイベントと該入力されたイベントのダイアログの遷移先となるダイアログとの対応関係を記述したテーブルとを含む画面遷移テーブルと、画面遷移テーブルに基づきダイアログ間の画面遷移を行うように制御する画面遷移制御手段とを備えるから、ネスティングフロー、シーケンシャルフローを含む画面遷移を効率良く実行することができる。

【0106】請求項2記載のマルチウィンドウシステムによれば、画面遷移テーブルが、複数回出現するようなモーダルダイアログに対して複数の遷移先を同一のリソースを利用して定義することが可能であるから、画面遷移を実現するためのリソースを必要最小限することができる、結果的にメモリなどの記憶手段に対する容量負荷を軽減することができる。

【0107】請求項3記載のマルチウィンドウシステムによれば、画面遷移テーブルが、複数のダイアログの遷移を1つにまとめてグループ化してグループ単位で記述可能であるから、画面遷移テーブルを記述する際に、同じ記述を重複して行う手間を省くことができ、また余分な記述の手間が省かれ、記述間違いを減らすことができ、可読性に優れた画面遷移テーブルを作成することができる。

【0108】請求項4記載のマルチウィンドウシステムによれば、画面遷移テーブルが、共有化が行われたダイアログおよびダイアロググループに対して、その共有された側のダイアログのユーザのイベントに対する遷移先をそれまでの遷移状況に応じて変更可能であるから、本来トレードオフの関係であるリソースの最小化、画面遷移テーブル記述の効率化と、きめ細かい遷移先の設定とを共存かさせることが可能になる。

【0109】請求項5記載のマルチウィンドウシステムによれば、さらに、画面遷移テーブルに従ってダイアログが破棄された場合に、該ダイアログの破棄を確認する破棄確認手段と、破棄確認手段によりダイアログの破棄が確認されたときに、該ダイアログの破棄を該ダイアログのオーナーウィンドウに通知する通知手段とを備えるから、画面遷移を行う際にそれに伴い保持されているステータスを変更することができる。また、アプリケーションが保持している条件によって、ダイアログの破棄を確認し、破棄を中止することが可能になる。

【0110】請求項6記載のマルチウィンドウシステムによれば、さらに、画面遷移テーブルを無視してユーザが設定した特定のダイアログを表示し、該特定のダイアログ上で入力されたユーザのイベントに対して画面遷移を行うことなく該ダイアログを破棄する破棄手段を備えるから、画面遷移テーブルに記述してある静的なデータのみを使用した画面遷移だけでなく、ユーザが画面を操作中に登録したダイアログへの画面遷移など、動的に遷移する画面を変更することも可能になる。

【0111】請求項7記載の画面遷移方法および請求項13記載の記憶媒体によれば、ネスティングフロー、シーケンシャルフローを含む画面遷移を効率良く実行することができる。

【0112】請求項8記載の画面遷移方法および請求項14記載の記憶媒体によれば、画面遷移を実現するためのリソースを必要最小限することができ、結果的にメモリなどの記憶手段に対する容量負荷を軽減することができる。

【0113】請求項9記載の画面遷移方法および請求項15記載の記憶媒体によれば、画面遷移テーブルを記述する際に、同じ記述を重複して行う手間を省くことができ、また余分な記述の手間が省かれ、記述間違いを減らすことができ、可読性に優れた画面遷移テーブルを作成することができる。

【0114】請求項10記載の画面遷移方法および請求項16記載の記憶媒体によれば、本来トレードオフの関係であるリソースの最小化、画面遷移テーブル記述の効率化と、きめ細かい遷移先の設定とを共存かさせることが可能になる。

【0115】請求項11記載の画面遷移方法および請求項17記載の記憶媒体によれば、画面遷移を行う際にそれに伴い保持されているステータスを変更することがで

きる。また、アプリケーションが保持している条件によって、ダイアログの破棄を確認し、破棄を中止することが可能になる。

【0116】請求項12記載の画面遷移方法および請求項18記載の記憶媒体によれば、画面遷移テーブルに記述してある静的なデータのみを使用した画面遷移だけでなく、ユーザが画面を操作中に登録したダイアログへの画面遷移など、動的に遷移する画面を変更することも可能になる。

【図面の簡単な説明】

【図1】本発明のマルチウィンドウシステムの実施の第1形態を構築するための情報処理装置の構成を示すブロック図である。

【図2】図1の情報処理装置上に構築されているマルチウィンドウにおける画面遷移制御を実現するための構成を示す概念図である。

【図3】図1の情報処理装置上に構築されているマルチウィンドウシステムにおける画面遷移の手順を示すフローチャートである。

【図4】図1の情報処理装置のマルチウィンドウシステムにおける画面遷移開始の手順を示す図である。

【図5】図1の情報処理装置のマルチウィンドウシステムにおける画面遷移の手順を示す図である。

【図6】通常のネスティングフローによる画面遷移を示す図である。

【図7】図1の情報処理装置のマルチウィンドウシステムにおけるネスティングフローによる画面遷移を示す図である。

【図8】図1の情報処理装置のマルチウィンドウシステムにおけるダイアログの生成、破棄に関する処理の手順を示すフローチャートである。

【図9】画面を遷移させる関数内部における処理手順を示すフローチャートである。

【図10】CALLBACKDATA構造体の構成を示す図である。

【図11】図1の情報処理装置のマルチウィンドウシステムに用いられている画面遷移テーブルの作成方法を説明するための図である。

【図12】図1の情報処理装置のマルチウィンドウシステムに用いられている画面遷移テーブルの作成方法を説明するための図である。

【図13】図1の情報処理装置のマルチウィンドウシステムに用いられている画面遷移テーブルの作成方法を説明するための図である。

【図14】図1の情報処理装置のマルチウィンドウシステムに用いられている画面遷移テーブルの作成方法を説明するための図である。

【図15】図1の情報処理装置のマルチウィンドウシステムに用いられている画面遷移テーブルの作成方法を説明するための図である。

【図16】図1の情報処理装置のマルチウィンドウシ

テムにおけるハイパーリンク機能を説明するための図である。

【図17】図1の情報処理装置のマルチウィンドウシステムにおけるハイパーリンク機能を説明するための図である。

【図18】図1の情報処理装置のマルチウィンドウシステムにおけるハイパーリンク機能を説明するための図である。

【図19】図1の情報処理装置のマルチウィンドウシステムにおけるシーケンシャルフローを説明するための図である。

【図20】図1の情報処理装置のマルチウィンドウシステムにおけるダイアログデータの共有例を示す図である。

【図21】ダイアログの共有における画面遷移の考え方を示す図である。

【図22】図1の情報処理装置のマルチウィンドウシステムにおける複数のダイアログを共有する例を示す階層構造図である。

【図23】図22のダイアログをグループ化した例の階層構造を示す図である。

【図24】図22のダイアログ12以下をグループ化した例の階層構造を示す図である。

【図25】複数のダイアログをグループ化した場合のグループ内のボタン押下に対する動作を示す図である。

【図26】グループの破棄例を示す図である。

【図27】ダイアログの破棄の一例を示す図である。

【図28】メッセージ通知機構の処理手順を示す図である。

【図29】図1の情報処理装置のマルチウィンドウシステムにおける動的に設定したダイアログの表示例を示す図である。

【図30】図1の情報処理装置のマルチウィンドウにおけるダイアログの階層関係と内部情報の保持方法とを示すための図である。

【図31】図30の内部情報を保持する構造体を示す図である。

【図32】図30の内部情報の推移を説明するための画面遷移例を示す図である。

【図33】内部情報の推移を表す図である。

【図34】内部情報の推移を表す図である。

【図35】内部情報の推移を表す図である。

【図36】内部情報の推移を表す図である。

【図37】マルチウィンドウシステムの画面遷移方法の一つであるネスティングフローによる画面遷移例を示す図である。

【図38】ネスティングフローの場合におけるウィンドウのトリートメント構造を示す図である。

【図39】マルチウィンドウシステムの画面遷移方法の他の一つであるシーケンシャルフローによる画面遷移例

を示す図である。

【図40】シーケンシャルフローによる画面遷移が発生する場合におけるウィンドウのトリー構造を示す図である。

【図41】ダイアログが単に消滅する画面遷移例を示す図である。

【図42】画面遷移を表すウィンドウ階層構造の一例を示す図である。

【符号の説明】

1 マルチウィンドウプログラム

* 1 a 画面情報保持部

1 b 画面遷移制御部

1 c 画面表示部

1 d 画面遷移テーブル

2 OS

3 CPU

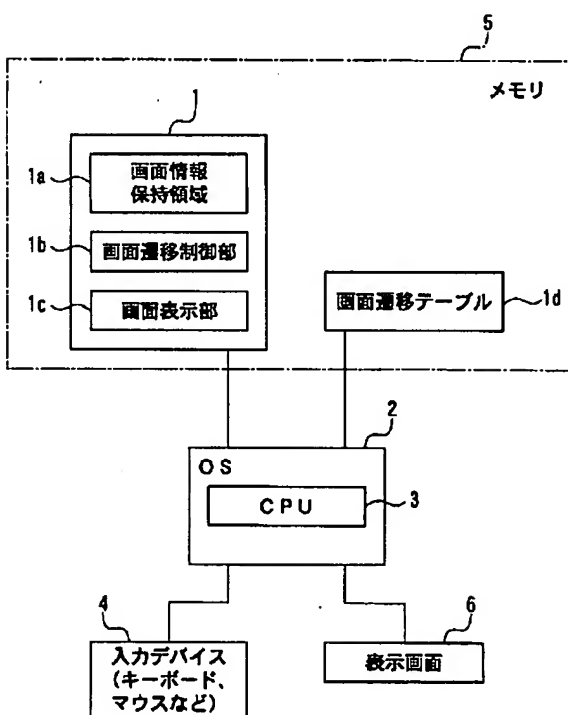
4 入力デバイス

5 メモリ

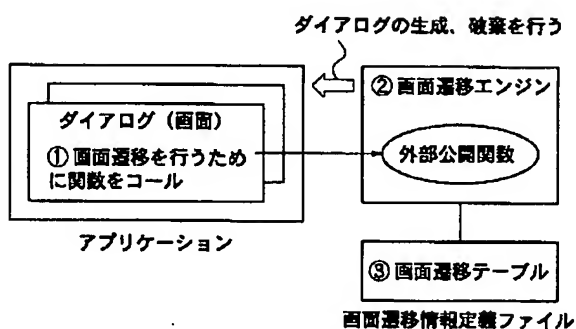
6 表示手段

*10

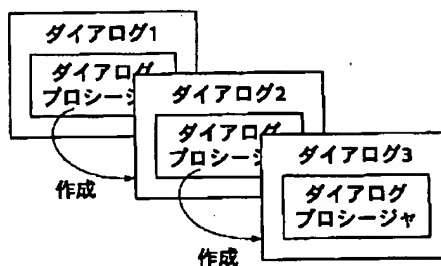
【図1】



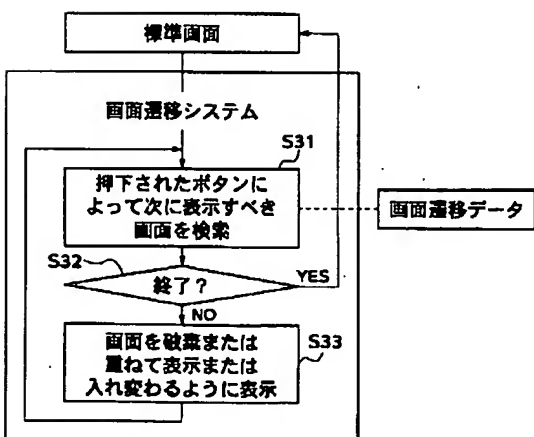
【図2】



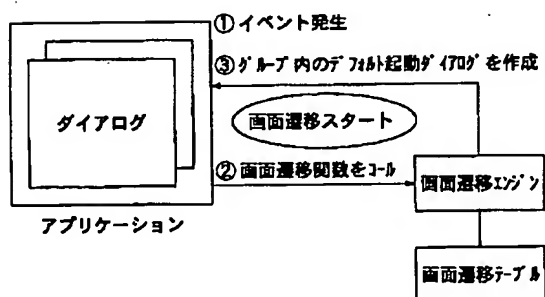
【図6】



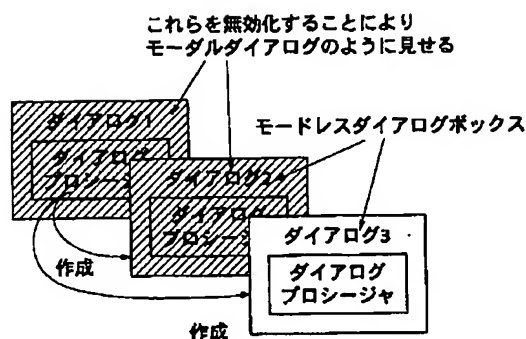
【図3】



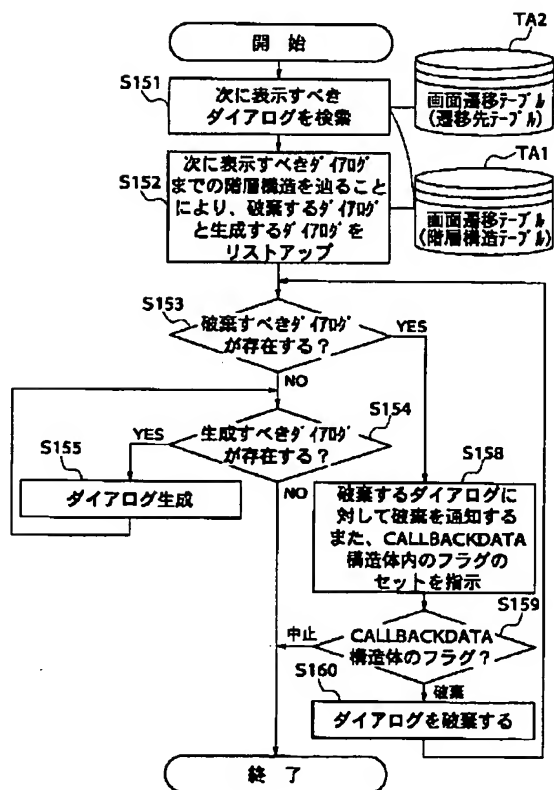
【図4】



【图7】



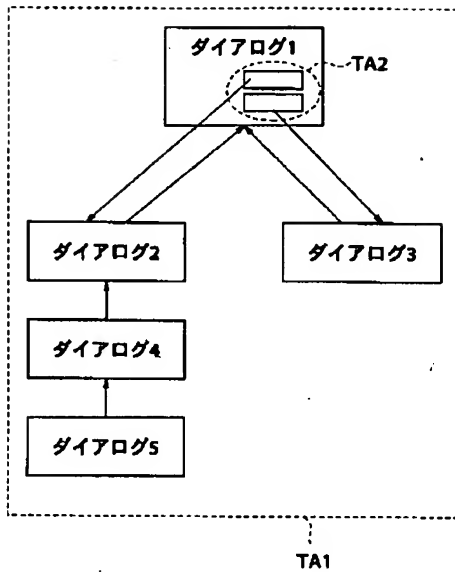
【图9】



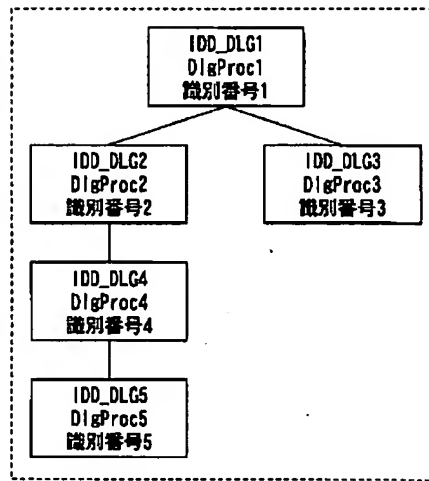
【图 10】

WORD CtrlID
WORD DlgID
HWND hWndParent
WORD ReturnID
BOOL InrptFlag

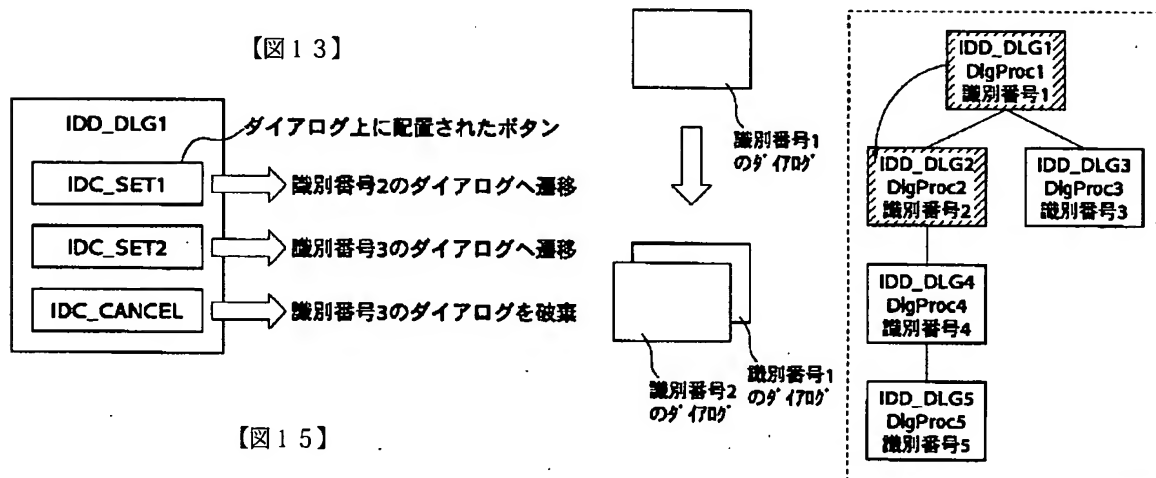
【図11】



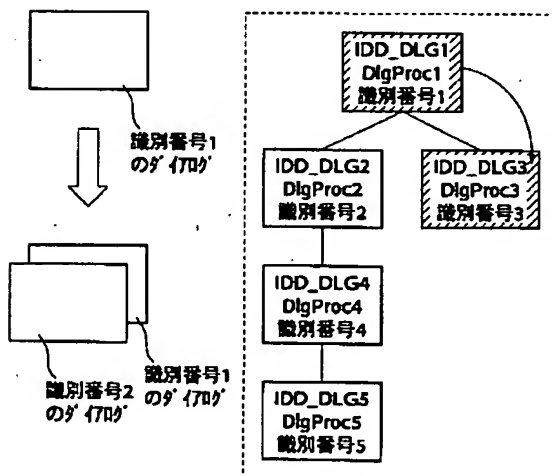
【図12】



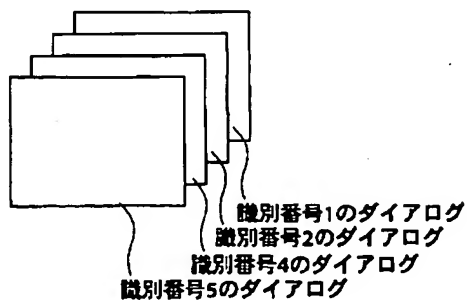
【図14】



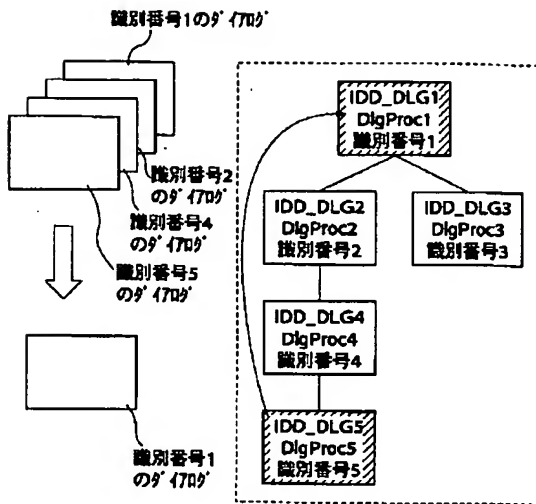
【図15】



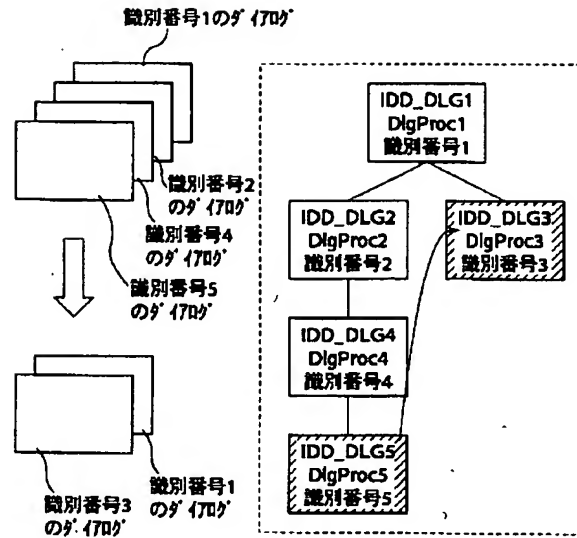
【図16】



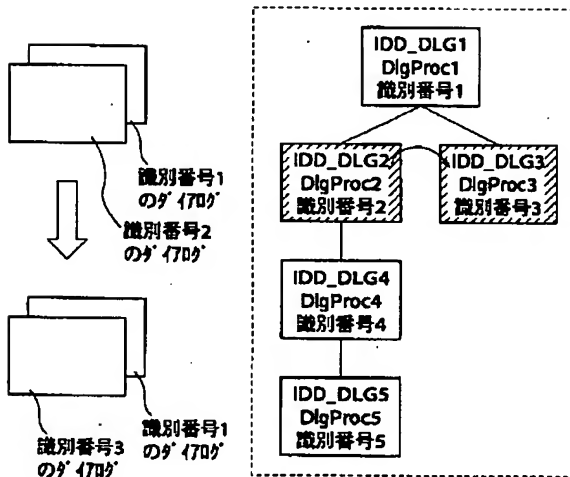
【図17】



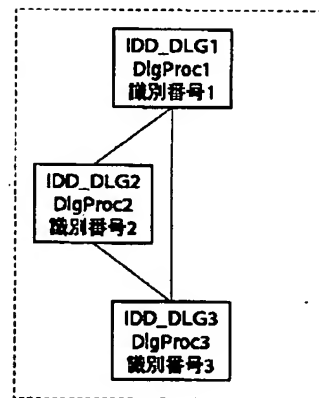
【図18】



【図19】

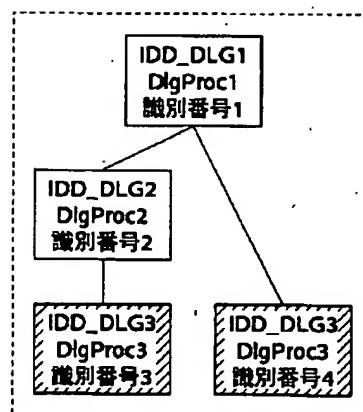
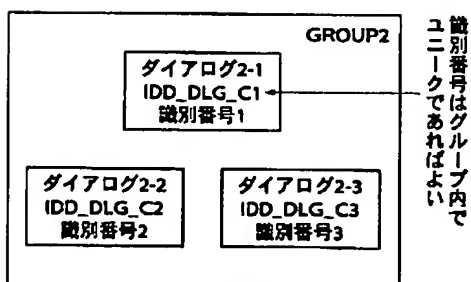


【図20】

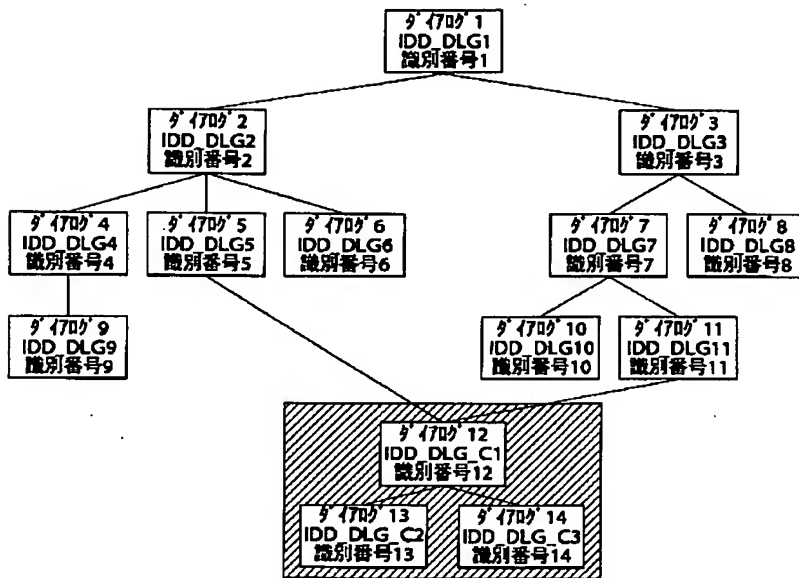


【図21】

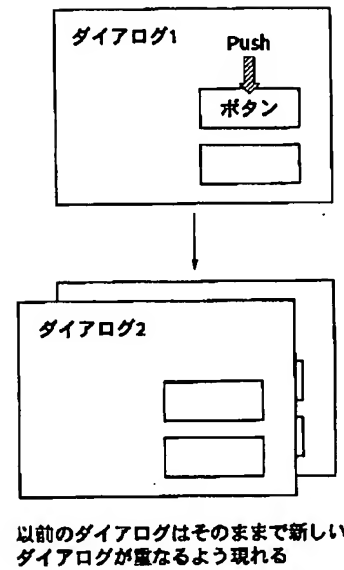
【図24】



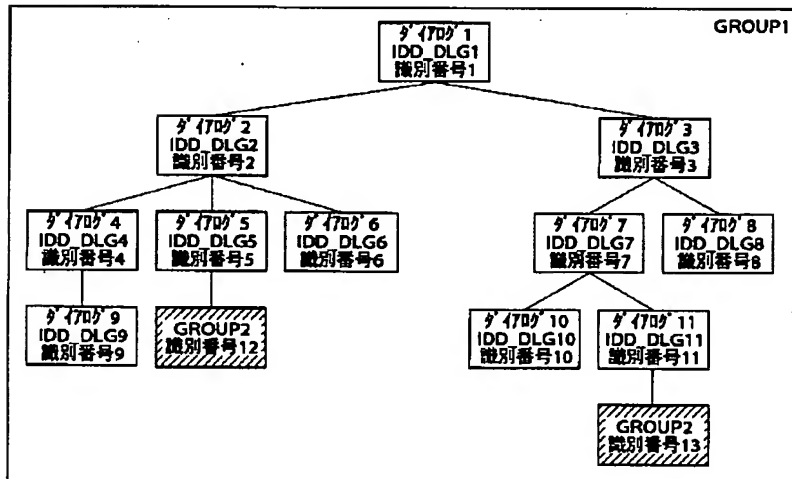
【図22】



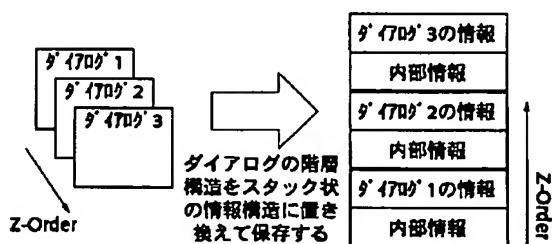
【図37】



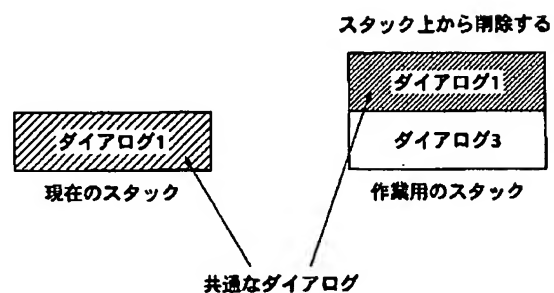
【図23】



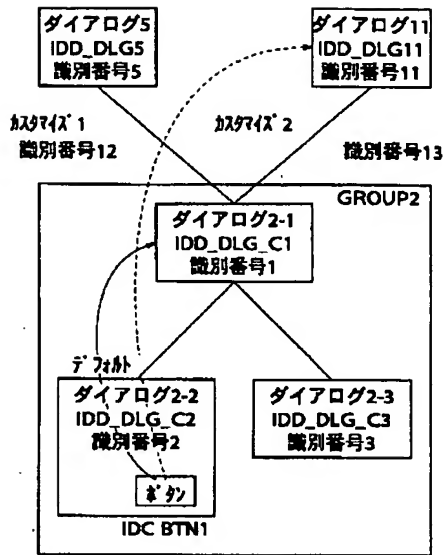
【図30】



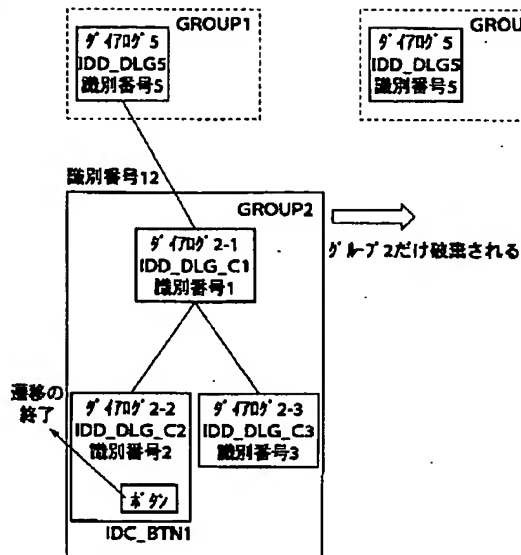
【図35】



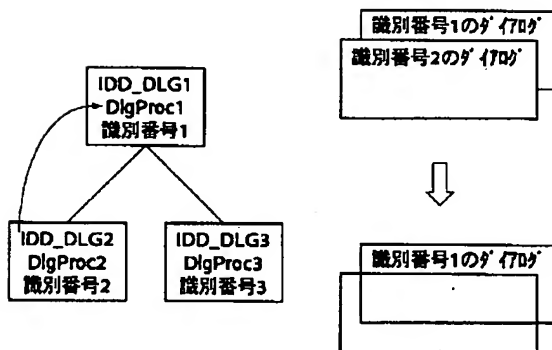
【図25】



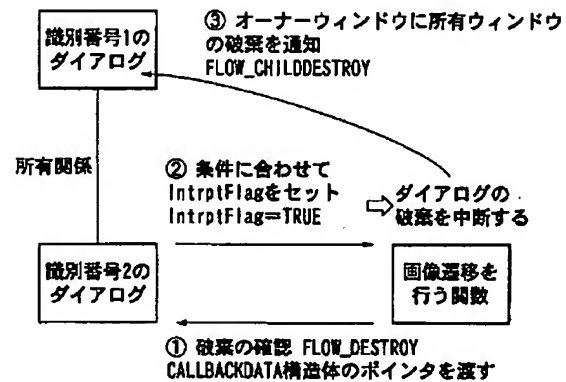
【図26】



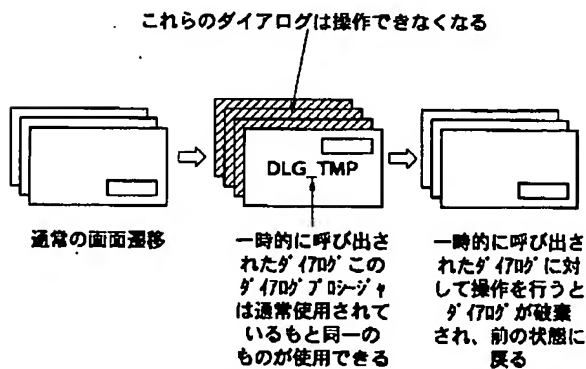
【図27】



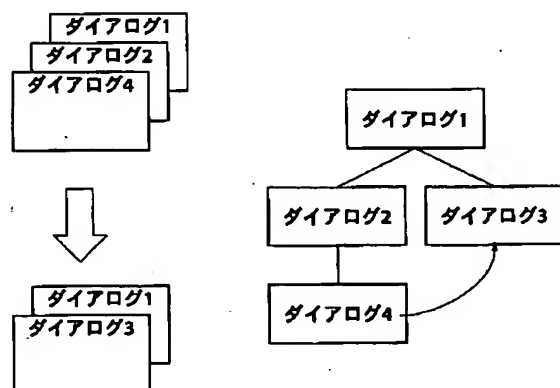
【図28】



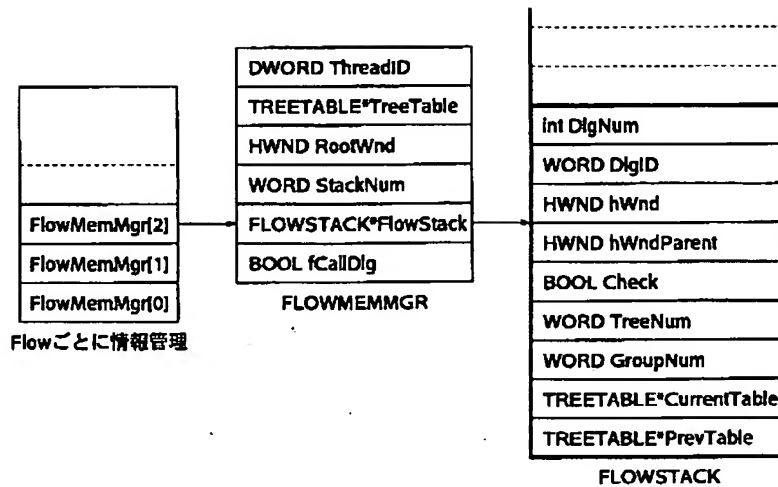
【図29】



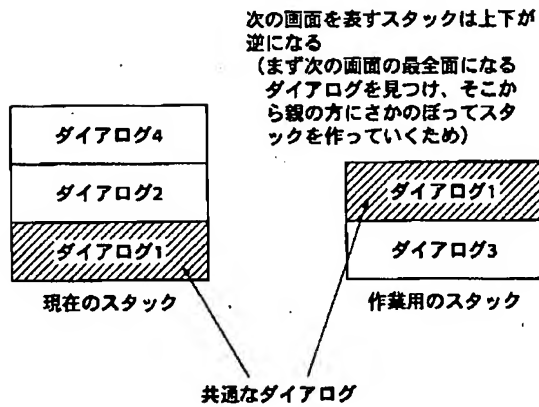
【図32】



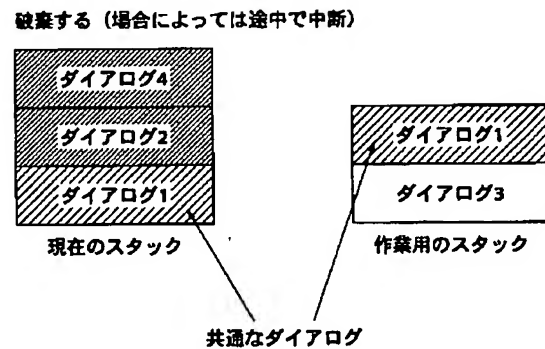
【図31】



【図33】

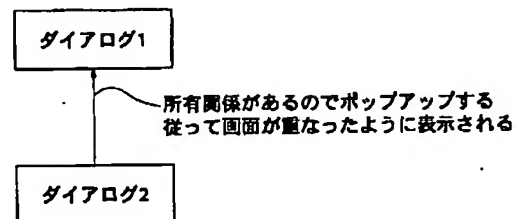
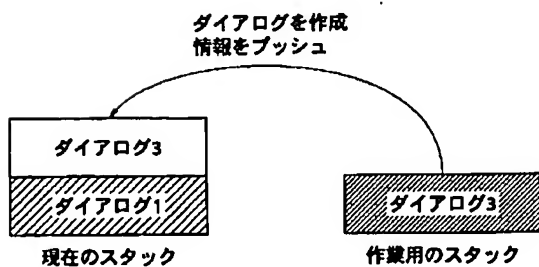


【図34】

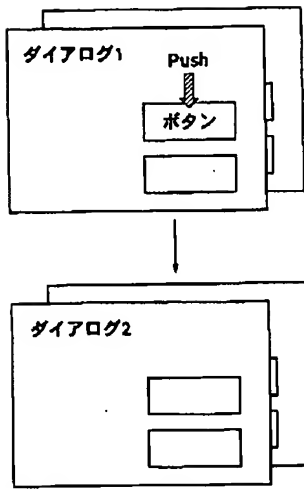


【図38】

【図36】

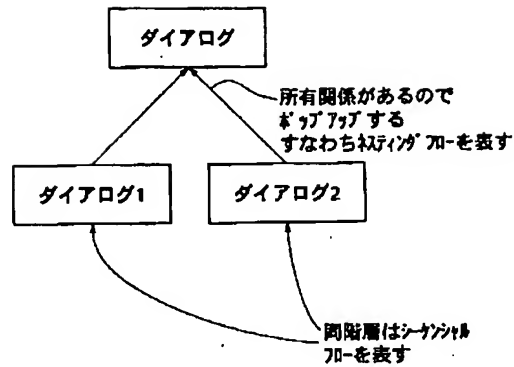


【図39】



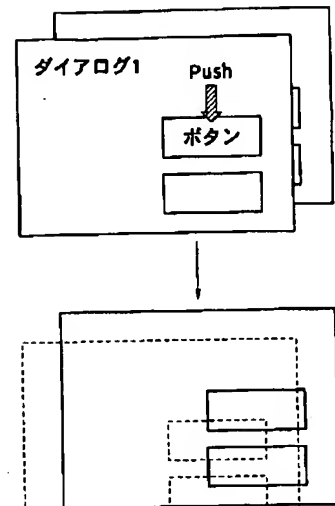
以前のダイアログが消えて新しい
ダイアログが現れる

【図40】



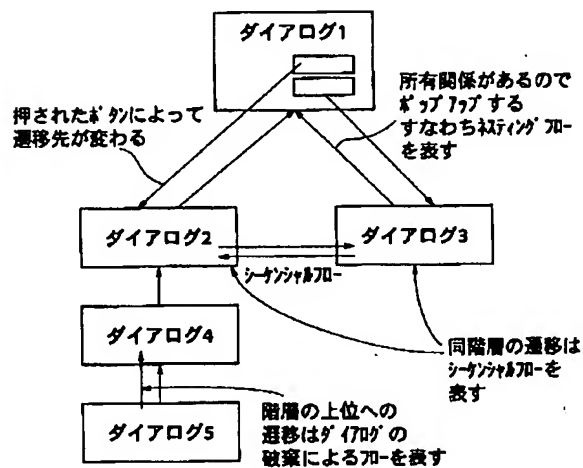
同階層はシグナル
70-を表す

【図41】



以前表示されていたダイアログ1
が消えた

【図42】



フロントページの続き

(72)発明者 佐々木 誠司
東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

Fターム(参考) 5E501 AA15 AC37 BA05 CA04 CB01
DA06 DA11 EA05 EA10 EA15
EB02 EB17 FA08 FA22 FB34